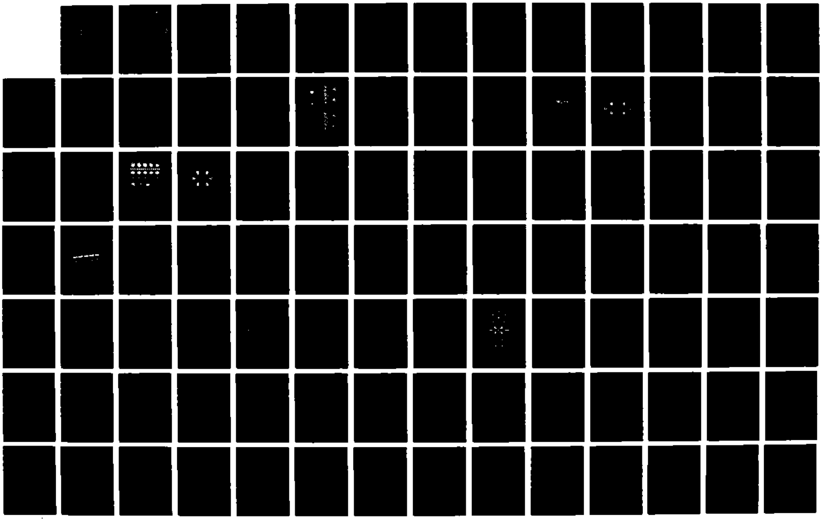
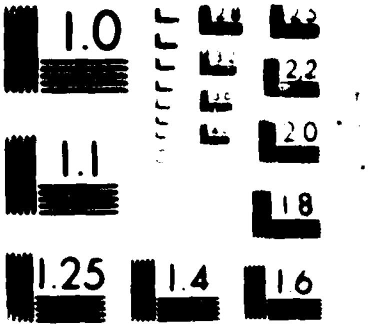
AD-A185 067 RELIABILITY MODELLING FOR FAULT-TOLERANT PARALLEL 1/2
PROCESSOR(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON
AFB OH E I GJERMUNDSEN JAN 87 AFIT/CI/NR/87-51T

UNCLASSIFIED F/G 12/6 NL




1.0
2.2
2.0
1.1
1.8
1.25
1.4
1.6

RELIABILITY MODELLING
FOR A
FAULT-TOLERANT PARALLEL PROCESSOR

by

Edvard I. Gjermundsen
Captain, United States Air Force

B.S., United States Air Force Academy
(1980)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE
DEGREE OF

MASTER OF SCIENCE IN
AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1987

Signature of Author Edvard I. Gjermundsen
Department of Aeronautics and Astronautics
January 1987

Certified by Jaynarayan H. Lala
Dr. Jaynarayan H. Lala
Technical Advisor

Certified by Wallace E Vander Velde
Professor Wallace E. Vander Velde
Thesis Advisor

Accepted by ____________________
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

RELIABILITY MODELLING

FOR A

FAULT-TOLERANT PARALLEL PROCESSOR

by

Edvard Ivar Gjermundsen

Submitted to the Department of Aeronautics and Astronautics
on January 16, 1987, in partial fulfillment of the
requirements for the Degree of Master of Science in
Aeronautics and Astronautics

ABSTRACT

An architectural configuration for a Fault Tolerant Parallel Processor (FTPP) is defined to meet a space system reliability and throughput requirement. The FTPP utilizes a flexible architecture that consists of a set of interconnected clusters, each of which consists of a set of interconnected processors. FTPP architectural and redundancy strategies are perturbed to define a more reliable system, and combinatorial reliability models are developed to analyze these perturbations. The perturbations examined include changes to the cluster architecture, the use of redundant clusters, the redistribution of tasks among clusters, and three cluster interconnection schemes: fully linked, centrally linked, and singly linked. The results of these analyses are used to define relationships between FTPP reliability, throughput, and architecture; which should be useful to a system-designer attempting to meet a specific application requirement. [illegible] ; [illegible] Control, Space Systems; Computer Applications). ←

Technical Advisor: Dr. Jaynarayan H. Lala

Title: Division Leader
Systems Architecture Division
Charles Stark Draper Laboratory

Thesis Supervisor: Dr. Wallace E. Vander Velde

Title: Professor of Aeronautics and Astronautics
Massachusetts Institute of Technology

ACKNOWLEDGEMENTS

I would like to express my gratitude to all the people who assisted me in this endeavor. Thanks to Dr. Jay Lala and Rick Harper for the benefit of their technical knowledge, and to Professor Vander Velde for his valuable time and great patience.

I also express my appreciation to the Air Force Institute of Technology for the opportunity to attend MIT, and to Dr. Dave Burke and the Draper Education Department for their generous and invaluable support throughout my stay in Boston.

Publication of this report does not constitute approval by the Draper Laboratory or the United States Air Force of the findings or conclusions contained herein.

Edward I. Gjermundsen

Edward I. Gjermundsen

TABLE OF CONTENTS

TABLE OF CONTENTS

(continued)

LIST OF FIGURES AND TABLES

LIST OF FIGURES AND TABLES
(continued)

LIST OF FIGURES AND TABLES
(continued)

51

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Real-time control of increasingly complex space systems requires the development of faster throughput computers. Examples of the throughputs of past and present computers are:

1. Apollo Guidance and Navigation Computer - 10 KIPS
2. Space Shuttle Data Management System - 400 KIPS [1]

(1 KIPS = one thousand instructions per second)

Examples of throughput requirements of projected systems are:

1. NASA Space Station Requirement - 15 MIPS
2. Advanced Military Spacecraft/Autonomous Interplanetary Spacecraft Requirements - 100+ MIPS [1]

(1 MIPS = one million instructions per second)

As spacecraft missions become more ambitious, breakthroughs in processor throughput technology will be necessary to meet the increasing demand for faster throughput computers. The throughput speeds of current state-of-the-art processors range from 3 to 5 MIPS, and it is doubtful that the high speed throughput requirements projected for future systems can be achieved using a single processor. In the event of a technological leap in processor throughput capability, designers will inevitably find the need for still faster throughputs. In addition to high speed throughput, processors for future spacecraft must possess a reliability commensurate with the real-time mission critical applications they control [2].

A solution to the projected gap between processing requirements and capabilities is parallel processing. Parallel processing is an efficient method of information processing that emphasizes the exploitation of concurrent events in the computing process by demanding the execution of many programs simultaneously [3].

Ideally, software partitioned into n pieces will run n times as fast in parallel; but, because of inefficient algorithms for exploiting concurrency in the computer problem and of the idling of the processor by conflicts over memory access or communication paths, in actual practice the speedup is much less [3]. According to Hwang and Briggs, estimates of the actual speedup range from a lower bound of $\log_2 n$ (Minsky's Conjecture) to an upper bound of $n/(\ln n)$, where n is the number of processors in the parallel system.

In order to meet the stringent reliability requirements associated with real-time mission critical applications, suitable architectures and redundancy strategies must be developed to exploit the redundancy inherent in parallel systems. Examining these architectures and redundancy strategies constitutes the focus of this thesis.

1.2 PURPOSE

This thesis seeks to contribute to the current effort at the Charles Stark Draper Laboratory to develop a high speed fault tolerant computer. The effort nas been designated as the Fault Tolerant Parallel Processor (FTPP). The purpose of this thesis is to define relationships among reliability, throughput, and computer architecture for the FTPP. This will be done in the context of applying requirements for a potential FTPP space system application. The FTPP must be capable of meeting application reliability and throughput requirements while minimizing FTPP hardware overhead and required FTPP component reliability.

It is recognized that determining absolute reliability figures through analysis lacks credibility without detailed knowledge of component behavior and past experience with similar designs not available in the early design phase. Analysis can, however, generate credible relative reliability figures that are required to define the desired relationships and to make decisions regarding suitable FTPP archictectures and redundancy strategies. Applying an actual reliability requirement also provides an appraisal of the suitability of the FTPP concept for one of its potential applications.

1.3 METHODOLOGY

The remainder of this thesis will define a baseline FTPP architecture and examine the effects of various modifications in order to optimally meet the application requirement. More specifically, the thesis will proceed as follows:

1. Define the general FTPP architecture and configuration constraints; and define FTPP component parameters to include Mean Time To Failure (MTTF), reconfiguration time, and throughput capability.

2. Define the space system application requirements. The application requirements will be reduced to required probability of success over a time period and required throughput.

3. Define a baseline FTPP architecture. The baseline FTPP architecture will be defined using the FTPP general architecture as a guide and be designed to meet the application throughput requirement.

4. Generate a FTPP reliability model and calculate the reliability of the baseline architecture.

5. Analyze the effects of intra-cluster modifications to the baseline architecture.

6. Analyze the effects of inter-cluster modifications to the baseline architecture.

7. Draw conclusions regarding the relationships among reliability, throughput, and computer architecture for the FTPP.

These steps will be addressed in the following chapters:

Chapter Two: Problem Description, addresses steps 1 through 7.

Chapter Three: Reliability Modelling, addresses steps 4 through 6.

Chapter Four: Conclusions/Recommendations, addresses step 7.

CHAPTER 2

PROBLEM DESCRIPTION

2.1 FAULT TOLERANT PARALLEL PROCESSOR DESCRIPTION [1,2,4]

The Draper Laboratory Fault Tolerant Parallel Processor (FTPP) is being designed to achieve high throughput and high reliability to meet the projected stringent requirements for future applications. To date, Draper Laboratory has defined a general architecture for the FTPP and begun component breadboarding.

The FTPP utilizes a flexible design concept where the architecture may be modified to suit the application. The FTPP consists of a set of building blocks arranged in clusters (figure 2.1-1).

2.1.1 Cluster Architecture

Each cluster within a FTPP consists of processor elements, network elements, input/output elements and memory elements.

PROCESSOR ELEMENTS (PE) - Processor elements perform the tasks of global controller, cluster controller and working processor element.

Global Controller (GC) - Manages inter-cluster communications; the loss of the GC implies system loss.

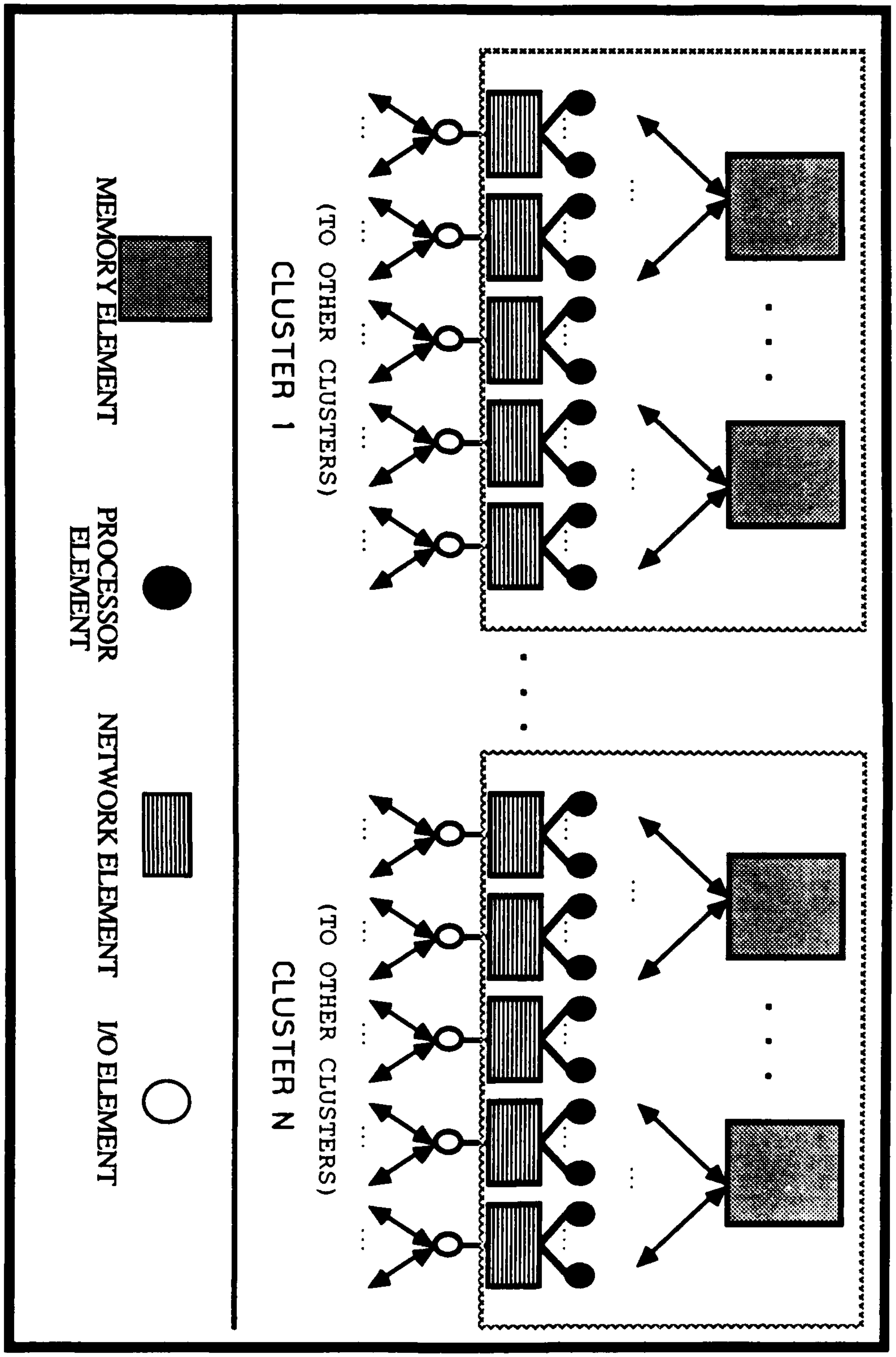


Figure 2.1-1: Fault Tolerant Parallel Processor (FTPP) General Architecture.

Cluster Controller (CC) - Manages intra-cluster communications; the loss of a CC implies cluster loss.

Working Processor Element (WPE) - Performs the computational application tasks; the loss of a WPE implies task loss.

Each processor element is assumed to have an exponential failure rate, a reconfiguration time averaging .25 seconds, and a throughput cabability of 5 MIPS.

NETWORK ELEMENTS (NE) - Network elements serve to pass information from processing elements to input/output elements, and are an integral part of the FTPP architecture. The loss of a network element results in the loss of all the processors connected with the network element and one input/output element. When there are three processor elements per network element, each network element is assumed to have an exponential failure rate with a MTTF 10 times greater than the MTTF of a processing element. As more processor elements are connected to a network element, its increased complexity results in an decreased reliability. In modelling the greater complexity of a network element where more than three processor elements are used, the failure rate for the network element is assumed to increase ten percent for each additional processor element added. Preliminary reliability modelling has suggested five network elements per cluster [5].

INPUT/OUTPUT ELEMENTS (IOE) - Input/output elements serve to pass information between clusters via communication lines. Input/output elements are critical in determining the topology of a group of clusters, and the loss of an input/output element implies the loss of a communication link. Each input/output element is assumed to have an exponential failure rate with a MTTF equal to the MTTF of a processing element. Each input/output element's failure rate is assumed to increase an additional 10 percent for each communication line stemming from it.

MEMORY ELEMENTS (ME) - Memory elements are classed as either global memory or regional memory.

Global Memory (GM) - Stores information germane to every processor element's perception of the system state, but which often changes as a result of modifications or of updates by the processor elements. Global memory is assigned to each cluster, and the loss of global memory implies cluster loss.

Regional Memory (RM) - Stores time-invariant information. Regional memory is assigned to n processor elements, and the loss of regional memory implies the loss of the processor elements assigned to the memory.

Each memory element is assumed to have an exponential failure with a MTTF equal to the MTTF of a processing element. Like network elements, the failure rate increases by 10 percent for each additional processor element over three.

2.1.2 Inter-Cluster Connectivity

A FTPP system consists of a set of interconnected clusters.

TOPOLOGY - Topology refers to the particular cluster interconnection scheme employed. Topology affects reliability, throughput, modularity, and maintainability. Cluster to cluster connections are depicted in figure 2.1-2. Each cluster to cluster 'link' consists of a set of five communications 'lines' between input/output elements. A more detailed discussion of topology is found in section 3.3.

2.1.3 System Failure

System failure occurs when the FTPP can no longer reliably meet the throughput requirement. More specifically, if no degradation of system throughput is permitted and there are no redundant clusters, system failure occurs when either of the following two conditions holds true:

1. Any cluster is declared failed. A cluster not declared operational is declared failed. For a cluster to be declared operational, at least one processor with access to working memory

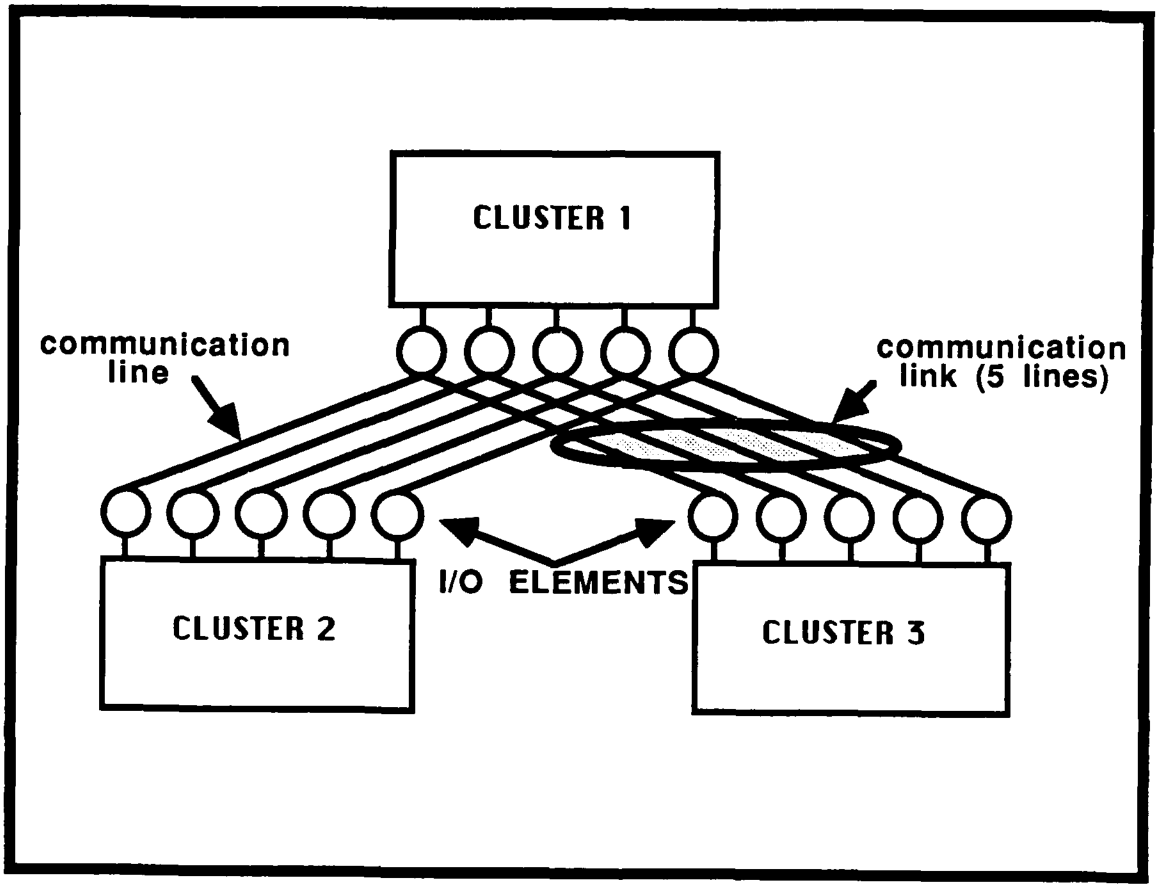


Figure 2.1-2: Cluster-to-Cluster Connectivity.

and network elements must be operational for each task assigned to the cluster; and at least two communications lines between the cluster and any other cluster must be operational. A communications line is declared operational if both input/output elements and their respective network elements are all operational.

2. Any cluster or group of clusters is isolated from the system. This requirement implies the system can be declared failed

even when there are no cluster failures and no individual cluster isolations. Figure 2.1-3 depicts a case where the communication links between Clusters 1 and 2 and between Clusters 4 and 5 have failed. Although there are no individual isolations, Clusters 1,5,6 are unable to communicate with Clusters 2,3,4 and the system is declared failed.

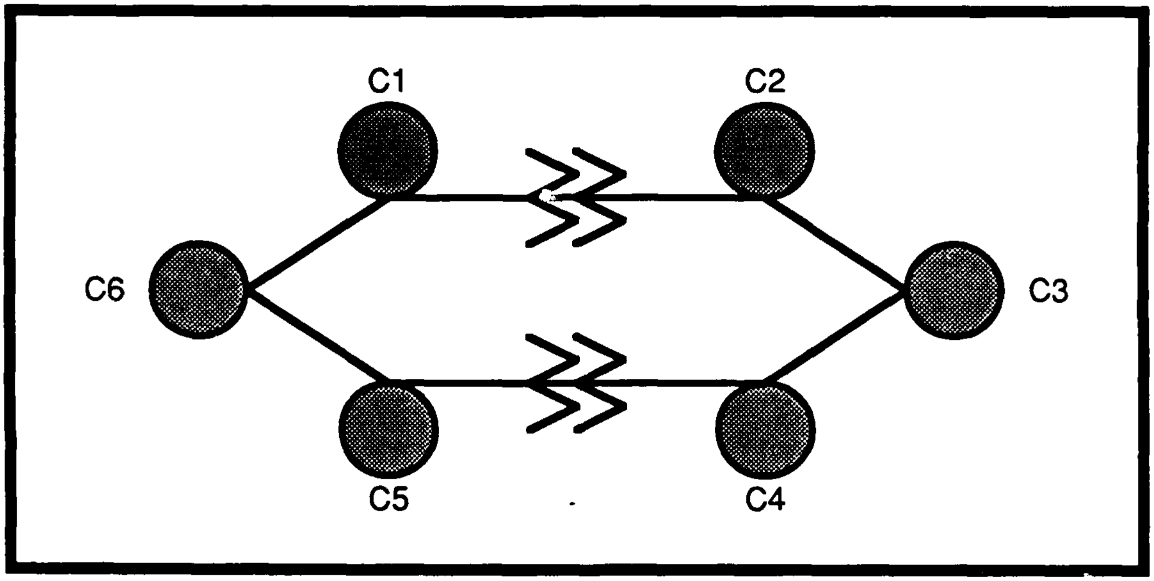


Figure 2.1-3: System Failure Due to Cluster Group Isolation.

2.2 APPLICATION SPECIFICATION

The specification to be examined by this thesis is stated as follows:

1. Out of 1000 spacecraft utilizing the FTPP, no more than one should be lost due to computer failure over a 20 year period.
2. The system will be maintained and fully repaired once per year.
3. The system requires a computational throughput of 100 MIPS.

The preceding specification defines the reliability requirements for various computer replication levels.

For a simplex computer system, the expected number of losses in 1 year will equal 1/20 or .05 spacecraft/year. Using the equation for the expected value of a binomially distributed variable yields:

$$E(L) = N(QS) \qquad (2.2\text{-}1)$$

where:

E(L) = Expected number of spacecaft losses due to computer failure in 1 year.

N = Initial number of operational spacecraft.

QS = Unreliability of the spacecraft computer system over 1 year.

Solving for QS yields an unreliability requirement of 5.0×10^{-5} over 1 year. Since the system consists of only a simplex computer, this requirement is the reliability requirement for that single computer.

For a duplex computer system: using combinatorial analysis and assuming a coverage of .85 for the first failure yields the equation:

$$QS = (1-C)P(1 \text{ computer fails})+P(2 \text{ computers fail}) \qquad (2.2\text{-}2)$$

where:

C = Coverage for a duplex computer system.

P(1 computer fails) = 2(1-Q)Q where Q is the unreliability of a single computer.

P(2 computers fail) = Q^2.

Solving for Q numerically yields an unreliability requirement of 1.661 x 10^{-4} over 1 year.

For a triplex computer system: using combinatorial analysis and assuming a coverage of 1.00 for the first failure and .85 for the second failure yields the equation:

$$QS = (1-C)P(2 \text{ computers fail})+P(3 \text{ computers fail}) \qquad (2.2\text{-}3)$$

where:

P(2 computers fail) = $3(1-Q)Q^2$.

P(3 computers fail) = Q^3.

Solving for Q numerically yields an unreliability requirement of 1.040 x 10^{-2} over 1 year.

For a quadruplex computer system: using combinatorial analysis and assuming a coverage of 1.00 for the first two failures and .85 for the third failure yields the equation:

$$QS = (1-C)P(3 \text{ computers fail})+P(4 \text{ computers fail}) \qquad (2.2\text{-}4)$$

where:

P(3 computers fail) = $4(1-Q)Q^3$.

P(4 computers fail) = Q^4.

Solving for Q numerically yields an unreliability requirement of 4.326 x 10^{-8} over 1 year.

In summary, the specification yields a computer unreliability requirement which varies with the computer replication level:

Computer Unreliability Requirement (Simplex Configuration) = 5.00 x 10^{-5}

Computer Unreliability Requirement (Duplex Configuration) = 1.66 x 10^{-4}

Computer Unreliability Requirement (Triplex Configuration) = 1.04 x 10^{-6}

Computer Unreliability Requirement (Quadruplex Configuration)= 4.32 x 10^{-8}

The reliability analyses in chapter 3 will utilize a range of component MTTFs. Processor, input/output, and memory elements possess equal MTTFs ranging from 10^4 to 10^5 hours. Network elements possess a MTTF ten times greater ranging from 10^5 to 10^6 hours. Figure 2.2-1 graphically depicts the equivalent reliability of cluster components when their MTTF's are taken over a one year period. The X-axis scale represents the MTTF of the processor, input/output, and memory elements and the MTTF of the network elements divided by ten. The X-axis scaling of MTTF will represent the same for all succeeding graphs in the thesis. Figure 2.2-1 may be used as a reference to translate the often used quantity of MTTF to the more meaningful quantity of reliability.

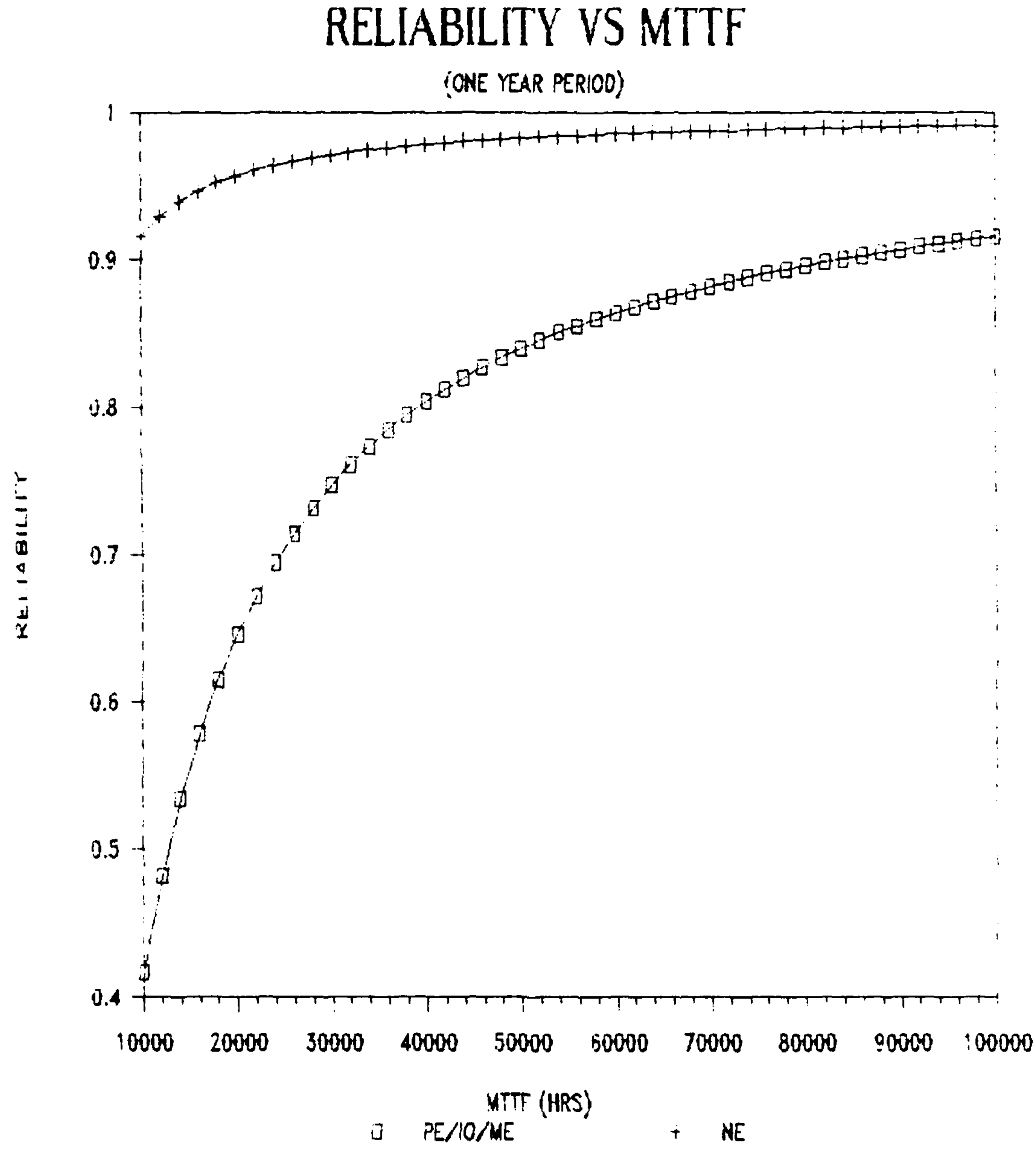


Figure 2.2-1: Component Reliability vs. Component MTTF.

2.3 BASELINE ARCHITECTURE

The following assumptions are made to define a baseline architecture:

1. There are three processor elements per network element.

2. There is one shared memory element for every three processor elements. Each memory element stores both global and regional memory.

3. Within a cluster, there are at least three processor elements assigned to a single task. There are at least four processor elements assigned to the global controller.

4. Clusters are fully linked (every cluster is connected to every other cluster).

5. A set of processor elements performing n tasks possesses a throughput n times as fast as a set of processors performing a single task (ideal behavior).

The baseline cluster architecture (figure 2.3-1) therefore utilizes:

1. 5 Input/Output Elements.

2. 5 Network Elements.

3. 15 Processor Elements.

   a. 12 Working Processor Elements. (The cluster designated global controller would utilize 8 working processor elements and 4 global controllers).

   b. 3 Cluster Controllers.

4. 5 Memory Elements.

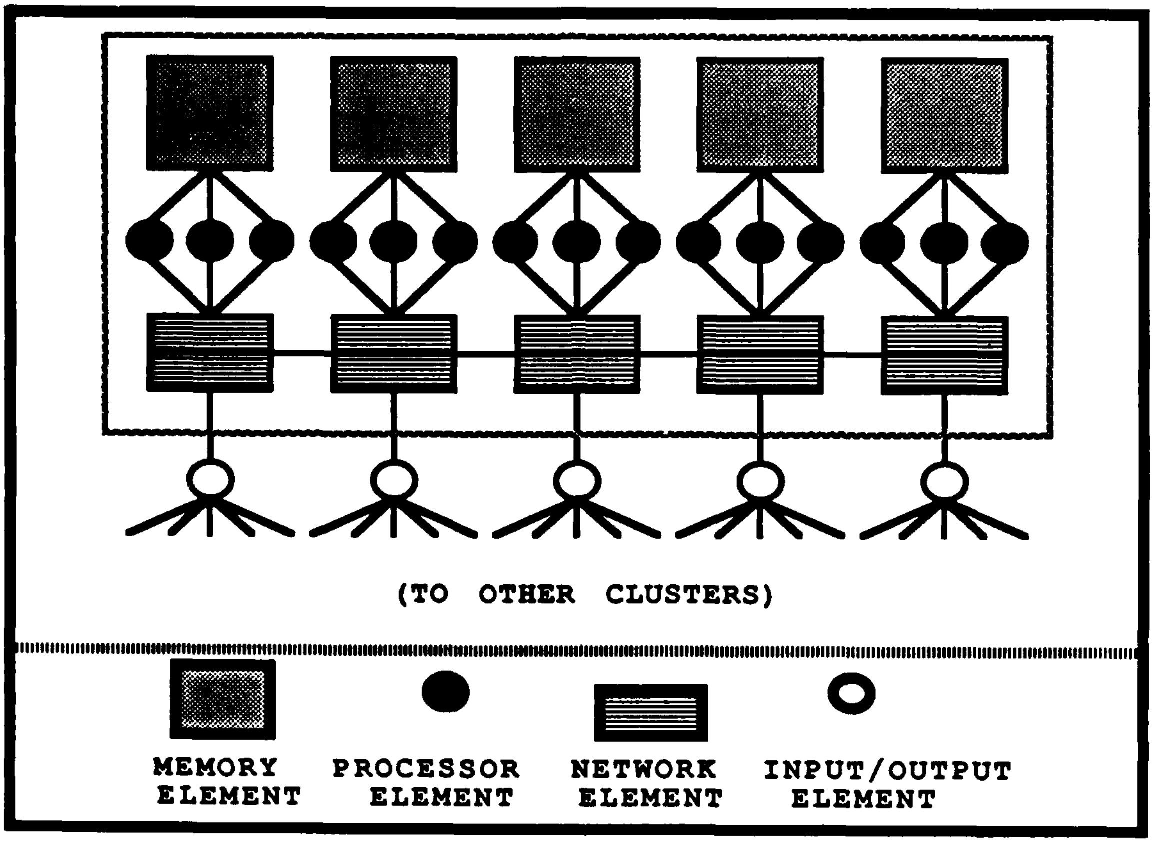


Figure 2.3-1: Baseline Cluster Architecture.

To support a 100 MIP requirement requires 20 sets of processors operating in parallel. A network of 6 baseline clusters (figure 2.3-2) will therefore meet the application throughput requirement leaving 8 processor elements as additional spares. The baseline system performs a total of 27 tasks: 20 computational tasks, 6 cluster controller tasks, and 1 global controller task.

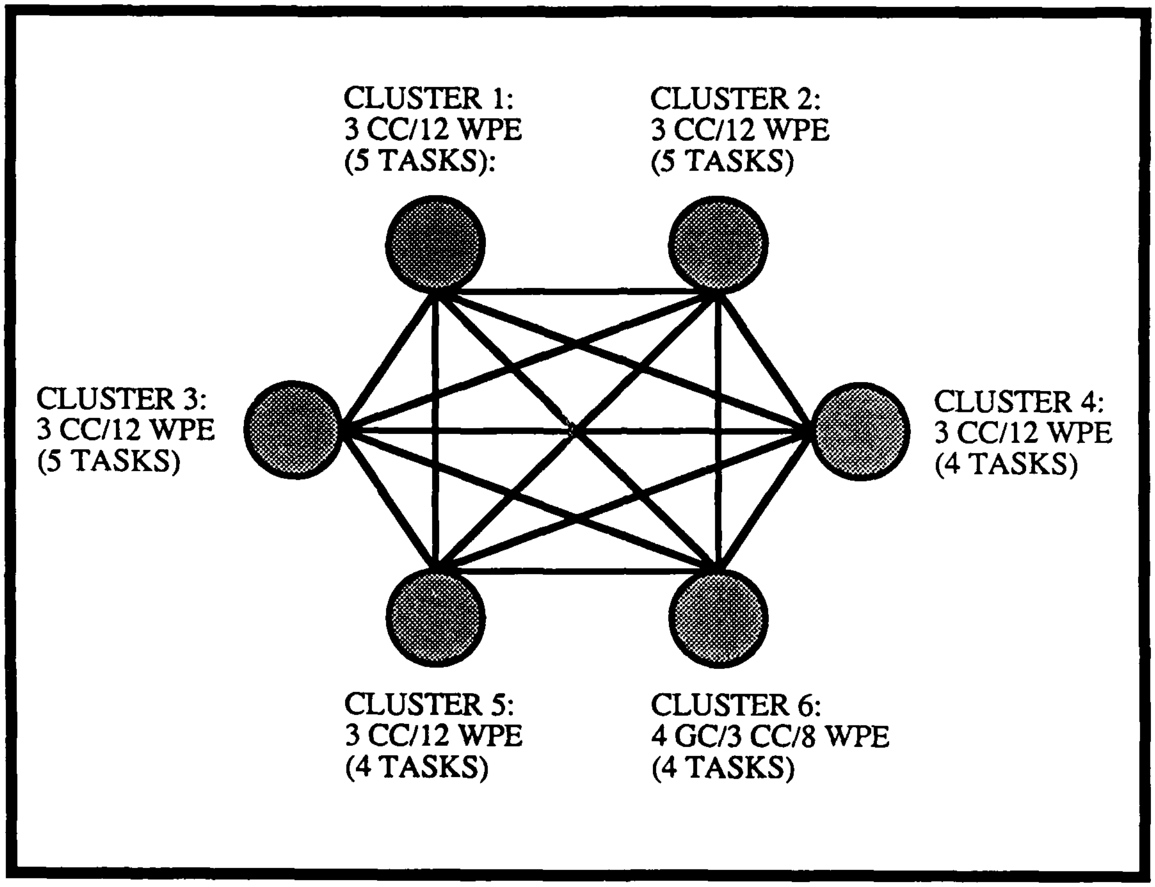


Figure 2.3-2: Baseline System Architecture.

CHAPTER 3

RELIABILITY MODELLING

3.1 BASELINE RELIABILITY

The purpose of this section is to develop a FTPP reliability model which generates the FTPP system reliability as a function of the FTPP component reliabilities. The reliability model will be used to generate an upper and a lower unreliability bound for the baseline architecture defined in Chapter 2. For the lower unreliability bound, any assumptions made generate the highest possible reliability. For the upper unreliability bound, any assumptions made generate the highest possible unreliability. The one exception to these rules is that cluster isolations due to combinations of failures between network elements and input/output elements of different clusters are neglected for both upper and lower bound calculations. This exception has little effect on reliability calculations, as will be demonstrated presently, and permits the clusters to be treated as independent units. Both upper and lower unreliability bounds will be generated using combinatorial and decomposition techniques.

3.1.1 Justification for Cluster Independence

The fully linked topology of the baseline architecture from the point of view of one cluster is depicted in figure 3.1-1. Intuitively, failure modes which would isolate a cluster would primarily involve the elements of that particular cluster. The isolation of a cluster due to failures in other clusters would require a seemingly unlikely combination of failures in all other clusters. This observation is applicable for a cluster connected to a 'significant' number of clusters and may not be particularly applicable for a ring topology where every cluster is connected to only two other clusters. The following analysis calculates the probability of cluster isolation, due to input/output element failures, in two ways. First: the probability that a single cluster

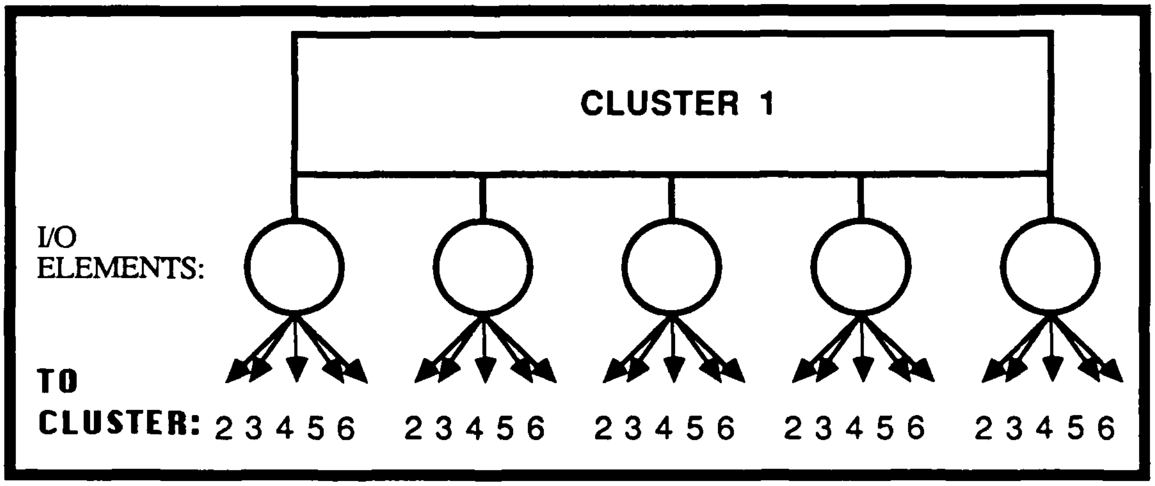


Figure 3.1-1: Baseline Architecture as Viewed from Cluster 1.

experiences an isolation is calculated neglecting isolations due to input/output element failures in other clusters and multiplying the reliabilities of the six clusters. Since some failure modes are neglected, the calculated unreliability will be lower than the actual. Second: the probability that a cluster experiences an isolation is calculated by taking into account all possible failure modes between clusters and multiplying the reliabilities of the six clusters. Since some failure modes will be counted more than once, the calculated unreliability will be higher than the actual. If the two methods yield similar results, the first method can be employed to simplify the system reliability model without significantly altering the results.

Using the first method: the probability of at least one isolation, P(ISOLATION), is the compliment of the probability of no isolations, P(0 ISOLATIONS):

$$P(\text{ISOLATION}) = 1-P(0\ \text{ISOLATIONS}) \qquad (3.1\text{-}1)$$

The probability of no isolations is calculated by multiplying the reliabilities of the six clusters:

$$P(0\ \text{ISOLATIONS}) = [1-P(1\ \text{CLUSTER ISOLATION})]^6 \qquad (3.1\text{-}2)$$

The probability a particular cluster experiences an isolation: P(1 CLUSTER ISOLATION), is the probability 4 or 5 input/output elements fail on that cluster since other cluster failures are being neglected:

$$P(1\ \text{CLUSTER ISOLATION}) = (1-\text{RIO})^5+5(1-\text{RIO})^4(\text{RIO}) \qquad (3.1\text{-}3)$$

where: RIO = Reliability of a single input/output element.

The probability of at least one cluster isolation is now completely defined as a function of the reliability of the input/output elements.

Using the second method: the probability of at least one isolation must take into account all failure modes. Equations 3.1-1 and 3.1-2 remain valid. To find the probability a particular cluster experiences an isolation, a decomposition on the input/output elements of that cluster is performed:

$$P(1\ \text{CLUSTER ISOLATION}) = \sum_{N=0}^{5}[P(1\ \text{CLUSTER ISOLATION}/N\ \text{IO WORK}) \times P(N\ \text{IO WORK})] \qquad (3.1\text{-}4)$$

where:

P(N IO WORK) = Probability exactly N out of 5 input/output elements operate.

P(1 CLUSTER ISOLATION/N IO WORK) = Probability a particular cluster is isolated given that exactly N input/output elements operate.

The probability exactly N input/output elements out of 5 operate can be calculated combinatorially:

$$P(N\ IO\ WORK) = \binom{5}{N}(1-RIO)^{5-N}(RIO)^{N}\ ;\ N=0\ \text{to}\ 5 \qquad (3.1\text{-}5)$$

where:

$\binom{5}{N}$ = Binomial coefficient calculated as 5!/[(5-N)! N!].

The probability a particular cluster is isolated given that exactly N input/output elements operate is calculated using the knowledge that a cluster must have at least two operational lines between itself and at least one other cluster. A single communication line between two clusters consists of 2 input/output elements, both of which must be operational for a communication line to be declared operational. Therefore:

P(1 CLUSTER ISOLATION/N IO WORK) =

P(N-1 or N out of N IO of all other clusters fail) (3.1-6)

where:

P(N-1 or N out of N IO of all other clusters fail)

$$= 1 \quad ; N=0,1$$

$$= \left(\sum_{i=0}^{1} \left[\binom{N}{i}(1-RIO)^{N-i}(RIO)^{i}\right]\right)^{5} \quad ; N=2 \text{ to } 5 \qquad (3.1\text{-}7)$$

The probability of at least one cluster isolation is now completely defined as a function of the reliability of the input/output elements.

The probability of at least one cluster isolation was programmed in FORTRAN using both methods. A comparison of the two methods for input/output element base MTTF's ranging from 10^4 to 10^5 hours is depicted graphically in figure 3.1-2. The base MTTF is the MTTF of an individual input/output element with no communication lines emanating from it. The graph shows a maximum error of about 2.3 percent occuring at input/output element MTTF=25000 hours and subsequently monotonically decreasing to about .1 percent for MTTF=100000 hours. The system reliability model generated in the following section will therefore assume that the probability that a particular cluster experiences an isolation is due exclusively to failures of its own elements. Clusters are in effect assumed independent for the fully linked system.

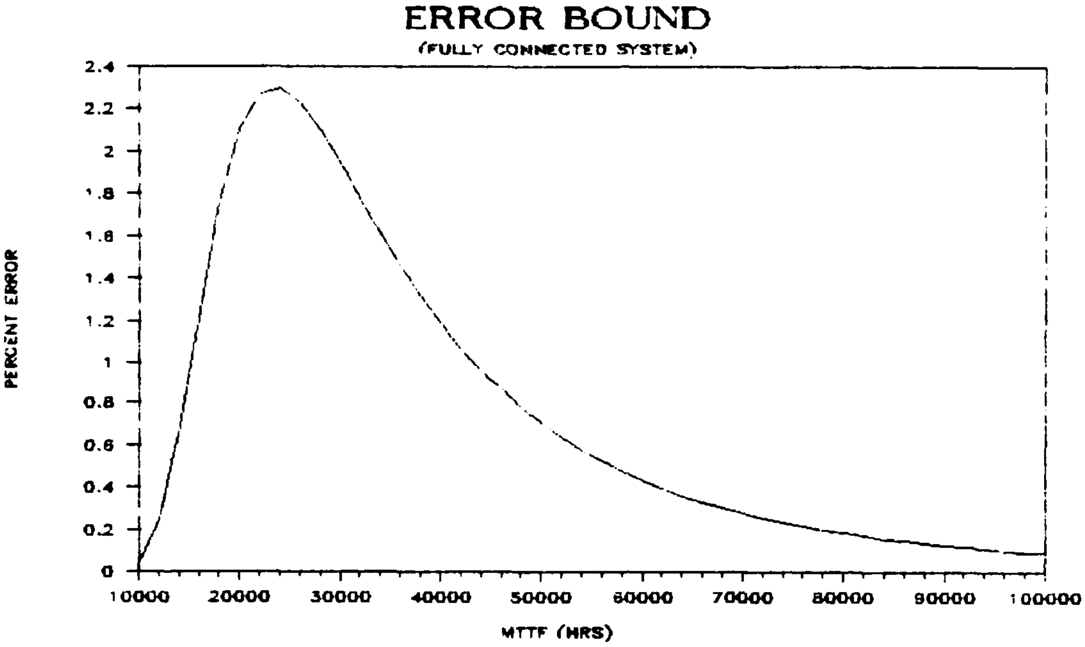


Figure 3.1-2: Percent Error Incurred Assuming Cluster Independence for Baseline System.

3.1.2 System Reliability Model for Baseline

The general procedure employed to derive a system reliability model is to find the reliability of a single cluster by decomposing on the network and memory elements. The reliability of the total system is found using the calculated cluster reliabilities and the assumption of cluster independence. The following derivation will calculate the system reliabilty in terms of the component reliabilities.

The probability that the FTPP system operates, P(S), is given by the following equation:

$$P(S) = P(C1)xP(C2/C1)xP(C3/C1 \cap C2)x \ldots xP(C6/C1 \cap C2 \ldots \cap C5) \tag{3.1-8}$$

where:

$P(CX) =$ Probability cluster X operates.

$P(CX/CA \cap \ldots \cap CE) =$ Probability cluster X operates given that clusters A through E operate.

Since cluster failures are assumed to be independent, the system reliability simply becomes the product of the individual cluster reliabilities:

$$P(S) = P(C1)xP(C2)xP(C3)xP(C4)xP(C5)xP(C6) \tag{3.1-9}$$

$$P(S) = \prod_{i=1}^{6} P(Ci) \tag{3.1-10}$$

To find the probability a cluster operates, P(C), a first decomposition is performed on the network elements:

$$P(C) = \sum_{X=0}^{5} [P(C/X\ NE\ WORK)xP(X\ NE\ WORK)] \tag{3.1-11}$$

where:

P(X NE WORK) = Probability exactly X out of 5 network elements operate.

P(C/X NE WORK) = Probability a cluster operates given that exactly X network elements operate.

P(X NE WORK) can be calculated combinatorially using equation 3.1-5 and substituting RN and X for RIO and N respectively where RN is the reliability of a single network element. The probability that a cluster operates, given that exactly X network elements operate, is equal to the probability input/output elements do not cause cluster failure, given that X network elements operate, and memory or processor elements do not cause cluster failure, given that X network elements operate. Since these two failure modes are independent, P(C/X NE WORK) is equal to the product of the reliability of the input/output elements, given that X network elements operate, and the reliability of the processor and memory elements, given that X network elements operate.

$$P(C/X\ NE\ WORK) = P(IO/X\ NE\ WORK) \times P(PE\ and\ ME/X\ NE\ WORK) \quad (3.1\text{-}12)$$

where:

P(IO/X NE WORK) = Probability input/output failures do not cause cluster failure given that X network elements operate.

P(PE and ME/X NE WORK) = Probability processor element or memory element failures do not cause cluster failure given that X network elements operate.

P(IO/X NE WORK) is the probability that at least two input/output elements have access to working network elements and can be can be calculated combinatorially:

$$P(IO/X\ NE\ WORK) = P(at\ least\ 2\ of\ X\ IO\ WORK) \quad (3.1\text{-}13)$$

where:

P(at least 2 of X IO WORK)

$$= 0 \quad ; X=0,1$$

$$= \sum_{i=0}^{x-2} [\binom{x}{i}(1-RIO)^{i}(RIO)^{x-i}] \quad ; X=2 \text{ to } 5 \quad (3.1\text{-}14)$$

where: RIO = Reliability of a single input/output element

P(PE and ME/X NE WORK) can be found by decomposing on the memory elements:

$$P(\text{PE and ME/X NE WORK}) = \sum_{Y=0}^{5} [P(\text{PE and ME/Y ME and X NE WORK}) \times P(\text{Y ME WORK})] \qquad (3.1\text{-}15)$$

where:

P(Y ME WORK) = Probability that exactly Y out of 5 memory elements operate.

P(PE and ME/Y ME and X NE WORK) = Probability that processor element or that memory element failures do not cause cluster loss given that Y memory elements and X network elements operate.

P(Y ME WORK) can be calculated combinatorially using equation 3.1-5 and substituting RM and Y for RIO and N respectively where RM is the reliability of a single memory element. To find the P(PE and ME/Y ME and X NE WORK) requires assumptions to be made regarding the behavior of the processor elements. For the lower unreliability bound, idealistic assumptions are made: processors within a cluster may perform any cluster task and switch tasks instantaneously with all failures covered. For the upper unreliability bound, pessimistic assumptions are made: each processor may perform only the function it was initially assigned, and failures are covered 85 percent of the time for duplex configurations and 100 percent of the time for triplex configurations and higher. These bounds were chosen so that practical systems fall between both bounds. At this point, the number of tasks a cluster is assigned to perform becomes an issue. In the baseline architecture, 3 clusters are assigned 4 tasks, and 3 clusters are assigned 5 tasks.

For the lower unreliability bound, at least 4 processors must operate for a 4 task cluster and at least 5 processors must operate for a 5 task cluster for the cluster to be declared operational. For a 4 task cluster: if all network elements and memory elements worked for the entire year, then 4 processors out of 15 must survive. If all network elements worked and 1 memory element failed during the year, then 3 processors are rendered useless and 4 out of the remaining 12 must survive. In some cases, the number of processors that must survive must be given in probabilistic terms. For example: if 1 network element and one memory element failed, 4 processors out of either 9 or 12 processors must survive depending upon which particular elements failed. In this case, 4 out of 12 must survive 20 percent of the time, and 4 out of 9 must survive 80 percent of the time. It is in this manner the following is computed:

P(PE and ME/(0 or 1 ME) or (0 or 1 NE) WORK) = 0

P(PE and ME/2 ME and 2 NE) = .1P(T of 6 PE WORK)

P(PE and ME/3 ME and 2 NE) = .3P(T of 6 PE WORK)

P(PE and ME/4 ME and 2 NE) = .6P(T of 6 PE WORK)

P(PE and ME/5 ME and 2 NE) = P(T of 6 PE WORK)

P(PE and ME/2 ME and 3 NE) = .3P(T of 6 PE WORK)

P(PE and ME/3 ME and 3 NE) = .6P(T of 6 PE WORK)+.1P(T of 9 PE WORK)

P(PE and ME/4 ME and 3 NE) = .6P(T of 6 PE WORK)+.4P(T of 9 PE WORK)

P(PE and ME/5 ME and 3 NE) = P(T of 9 PE WORK)

P(PE and ME/2 ME and 4 NE) = .6P(T of 6 PE WORK)

P(PE and ME/3 ME and 4 NE) = .6P(T of 6 PE WORK)+.4P(T of 9 PE WORK)

P(PE and ME/4 ME and 4 NE) = .8P(T of 9 PE WORK)+.2P(T of 12 PE WORK)

P(PE and ME/5 ME and 4 NE) = P(T of 12 PE WORK)

P(PE and ME/2 ME and 5 NE) = P(T of 6 PE WORK)

P(PE and ME/3 ME and 5 NE) = P(T of 9 PE WORK)

P(PE and ME/4 ME and 5 NE) = P(T of 12 PE WORK)

P(PE and ME/5 ME and 5 NE) = P(T of 15 PE WORK) (3.1-16)

where:

T = Number of tasks a cluster is assigned.

P(T of Z PE WORK) = Probability at least T processors out of Z operate.

P(T of Z) can be calculated combinatorially:

$$P(\text{T of Z PE WORK}) = 1-[\sum_{i=0}^{T-1}(\binom{Z}{i}(1-RP)^{Z-i}(RP)^{i})]$$

where: RP = Reliability of a single processor element.

For the upper unreliability bound, at least 3 processors must operate for a 3 task cluster and at least 4 processors must operate for a 4 task cluster. Since upper bound calculations assume processors perform only the functions they were initially assigned, a 4 task cluster would delegate tasks as depicted in figure 3.1-3. Each task is distributed

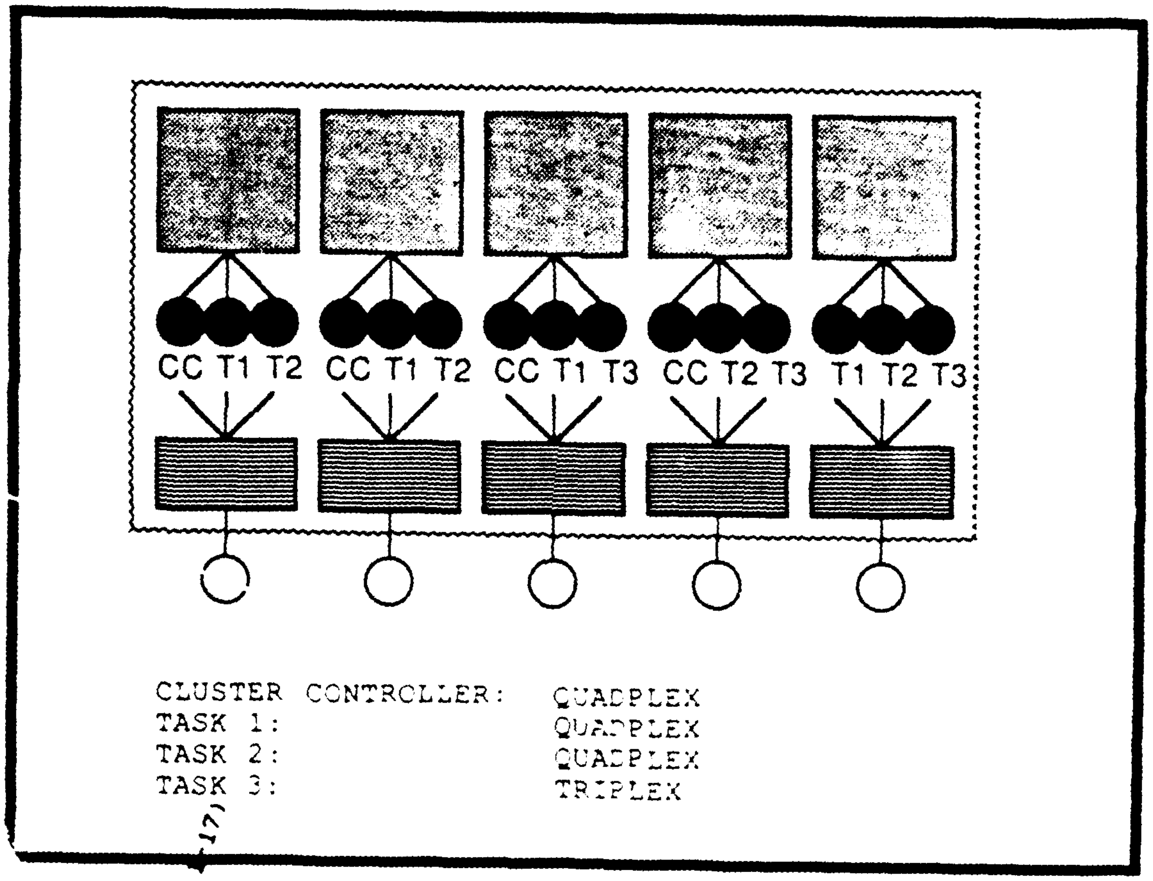


Figure 3.1-3: Delegation of Processors for a 4 Task Cluster.

among different network and memory elements to avoid single point failures. Three tasks would be controlled by quadruplexes of processors and one task would be controlled by a triplex of processors. If all network elements and memory elements worked for the entire year, then 1 processor from each quadruplex and 1 processor from the triplex must survive. As with the lower bound, some cases must be described probabilistically. If one network element and 1 memory element failed during the year, they are assumed to have failed at the beginning of the year. This assumption simplifies the model and tends to make the calculated unreliability higher than the actual reliability, thus maintaining the integrity of the upper bound. In this case, 1 triplex of processors and 3 duplexes of processors would remain in 56 percent of the cases. Two triplexes, 1 duplex, and 1 simplex would remain in 24 percent of the cases. One quadruplex, 2 triplexes and 1 duplex would remain in 12 percent of the cases and 4 triplexes would remain in 8 percent of the cases. The P(PE and ME/Y ME and X ME) equations (3.1-16) derived for the lower bound hold true, but P(T of X PE) is redefined as follows:

P(4 of 6 PE) = .9(RD^2 x RS^2)

P(4 of 9 PE) = .3(RT^2 x RD x RS) + .7(RT x RD^3)

P(4 of 12 PE) = .6(RQ x RT^2 x RD) + .4RT^4

P(4 of 15 PE) = RQ^3 x RT

P(5 of 6 PE) = .5(RD x RS^4)

P(5 of 9 PE) = .5(RT x RD^2 x RS^2) + .5(RD^4 x RS)

P(5 of 12 PE) = RT^2 x RD^3

P(5 of 15 PE) = RT^5 (3.1-18)

where:

RS = the reliability of a single processor:

$$RS = RP \qquad (3.1\text{-}19)$$

RD = the reliability a duplex of processors:

$$RD = 1-[(1-RP)^{2}+ (1-C)2RP(1-RP)\ ] \qquad (3.1\text{-}20)$$

RT = the reliability of a triplex of processors:

$$RT = 1-[(1-RP)^{3}+ (1-C)3RP(1-RP)^{2}] \qquad (3.1\text{-}21)$$

RQ = the reliability of a quadruplex of processors:

$$RQ = 1-[(1-RP)^{4}+ (1-C)4RP(1-RP)^{3}] \qquad (3.1\text{-}22)$$

C = Coverage of duplex of processors = .85

The reliability models for both upper and lower bounds were programmed using FORTRAN. The baseline FTPP unreliability as a function of component MTTFs is depicted graphically in figure 3.1-4. These bounds are compared to the required unreliability bounds derived in section 2.2 for various FTPP redundancy levels. Clearly, the baseline architecture is undesirable in terms of the application reliability requirement. Only the most idealistic processor behavior (lower bound assumptions) and optimistic MTTFs would support even a quadruplex computer system. Also note that the baseline unreliability bounds are not straight lines as would be expected in a system with an exponentially distributed failure time since the graph plots the log of unreliability. Since there is parallelism involved in the architecture, the total system need not be exponentially distributed, even though all components of the FTPP are assumed to be exponentially distributed [6]. After MTTF of approximately

35000 hours, hovever, both curves are relatively straight and could be approximated exponentially. The following sections in this chapter explore methods to further lower the unreliability curve.

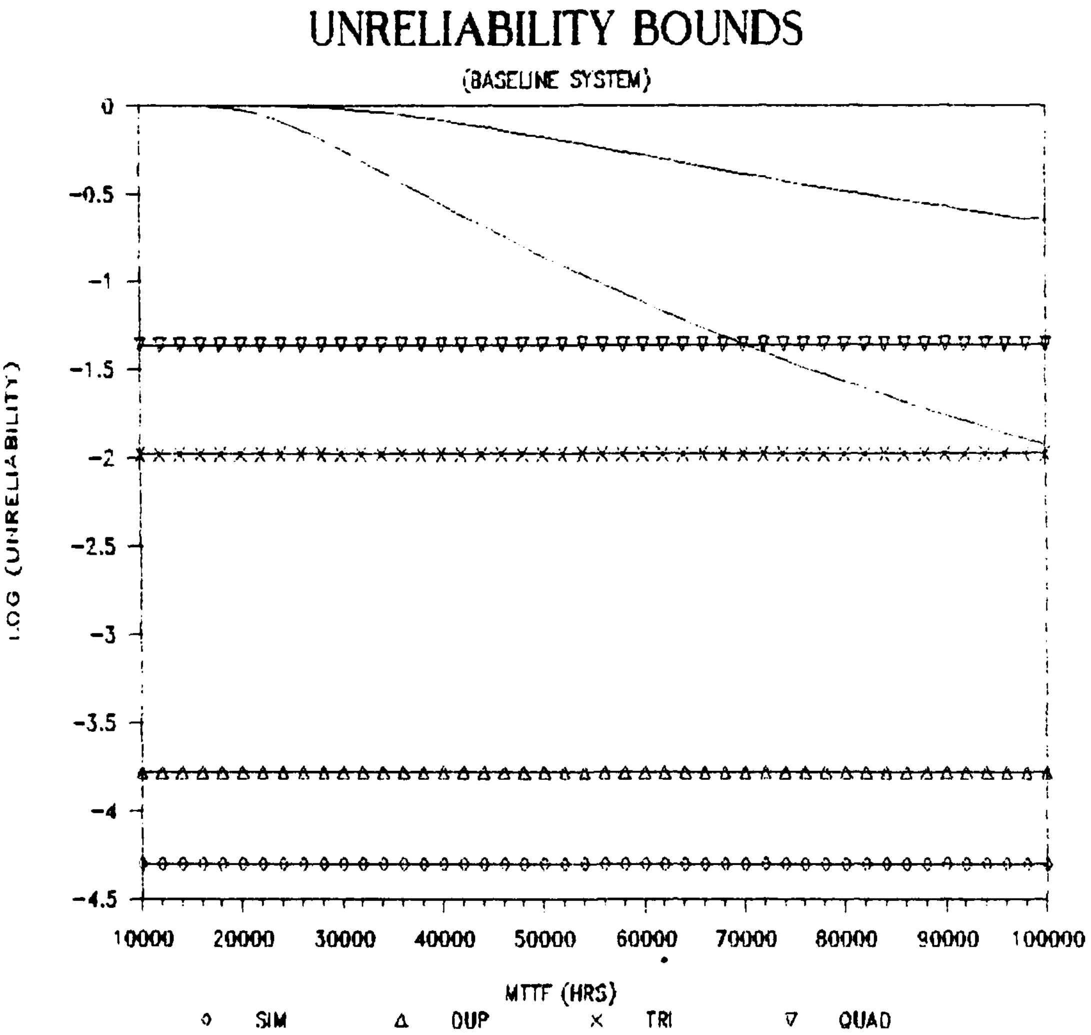


Figure 3.1-4: Baseline System Lower and Upper Unreliability Bounds.

3.2 INTRA-CLUSTER MODIFICATION OPTIONS

This section will address modifications at the cluster level and their effect on cluster reliability. Section 3.2.1 examines the sensitivity of the cluster reliability to changes in component MTTF. The remainder of the sections focus primarily on the processor elements which represent the most flexible components of the FTPP in terms of architectural and redundancy strategy modification. Section 3.2.2 examines modification of the number of processor elements per network and memory element. Section 3.2.3 examines the implications of assigning different numbers of tasks to the processors of a cluster. Section 3.2.4 examines the trade offs between reliability and throughput inherent in parallel processors and section 3.2.5 examines reliability bottlenecks.

3.2.1 Cluster Reliability Sensitivities

Cluster reliability sensitivities determine which component improvements provide the highest payoff. Using the reliability model developed in section 3.1, the MTTF of the four components was set constant at MTTF=50000 hours. Each element's MTTF was then varied individually from 10000 to 100000 hours. The results of this exercise using both lower and upper unreliability assumptions on a baseline cluster are depicted graphically in figures 3.2-1 and 3.2-2 respectively. Using lower bound

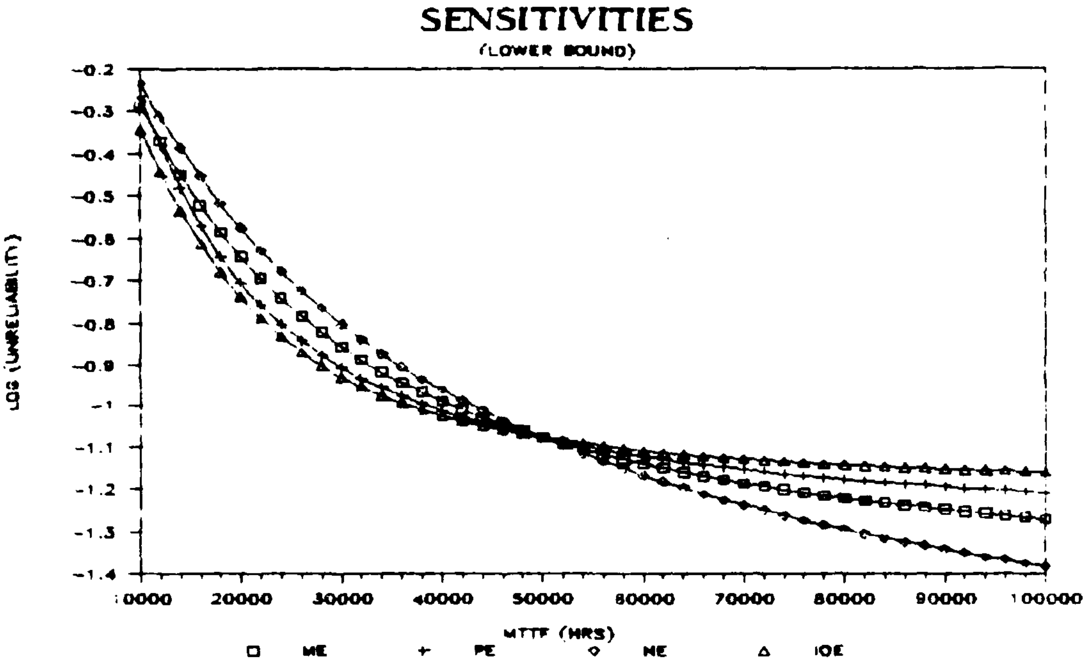


Figure 3.2-1: Cluster Reliability Sensitivities (Lower Bound Assumptions).

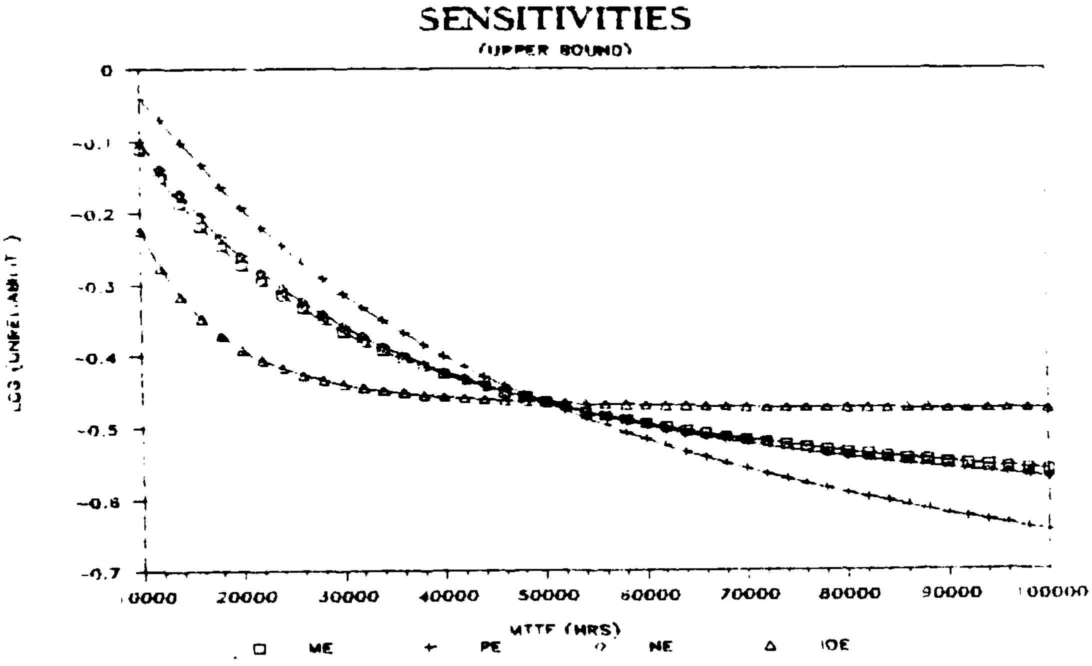


Figure 3.2-2: Cluster Reliability Sensitivities (Upper Bound Assumptions).

assumptions, cluster reliability is most sensitive to changes in the MTTF of network elements followed by memory, processor, and input/output elements. Using upper bound assumptions, cluster reliability is most sensitive to changes in the MTTF of processor elements followed by network, memory, and input/output elements.

In both lower and upper bound cases: network elements figure prominently, and input/output elements figure least prominently in determining cluster reliability. Loss of a network element renders three processor elements, one memory element, and one input/output element useless. Because of the dispersion of processor tasks discussed earlier, the loss any two network elements will never by itself cause cluster failure. Loss of an input/output element, on the other hand, has no effect on the usefulness of other cluster components. Using upper bound assumptions, processor elements jump to most prominent status in determining cluster reliability. Using upper bound assumptions, the loss of only two processors may cause task loss and, therefore, cluster failure. In both cases, memory elements fall in the middle. Loss of a memory element renders three processor elements useless. Once again, because of the dispersion of processor tasks, the loss of any two memory elements will never by itself cause cluster failure.

3.2.2 Cluster Architecture

The baseline cluster architecture utilized three processors for each network and memory element, and the general FTPP architecture allows modification of this number. Modifying the number of processors results in two opposing effects on cluster reliability. As more processors are added to a cluster, the reliability of the processors to perform the assigned tasks increases. In order to support this modification, the complexity of the network and memory elements increases, thus decreasing their reliablility. Whichever effect dominates under the existing conditions determines the optimum number of processors per network and memory element. To examine these effects analytically, consider a system consisting of one memory element, one network element, and N processor elements where N is the number of processors per network and memory element. The reliability of this trio of elements (RT) is the product of their reliabilities because all three must operate for a single task to be accomplished:

$$RT = RME_N \times RNE_N \times RPE_N \qquad (3.2\text{-}1)$$

where:

RME_N = The reliability of a memory element with N connections and failure rate L.

$$RME_N = e^{-(L+.1LN)t} \qquad (3.2\text{-}2)$$

RNE_N = The reliability of a network element with N connections and failure rate .1L.

$$RNE_N = e^{-(.1L+.01NL)t} \qquad (3.2\text{-}3)$$

RPE_N = The reliability of a set of N processors, each with failure rate L, to perform a task. The unreliability of each processor (Q) equals $1-e^{-Lt}$. The coverage of a duplex of processors is C and the coverage of triplexes and above is 1.

$$RPE_N = 1-[(Q)^N + N(1-C)(Q)^{N-1}(1-Q)] \qquad (3.2\text{-}4)$$

The optimum N is the N which maximizes the reliability of the trio. This N can be computed by taking the derivative of RT with respect to N, setting the result equal to zero, and solving for N:

$$\frac{d(RT)}{dN} = X - Q^N [X + \ln(Q) + Q^{-1}(1-Q)(1-C)(1+XN+N\ln(Q))] \qquad (3.2\text{-}5)$$

where: $X = -.11Lt$

Equation 3.2-5 can not be solved for N explicitly. The optimum number of processors was derived numerically for C=.85 and t=8760 hours. Results are depicted graphically in figure 3.2-3 for a range of component MTTFs. The optimum number of processor elements per network and memory element decreases as component MTTF increases. For the values chosen, the optimum number of processors is four for component MTTFs below 20000 hours (200000 hours for network elements) and three for MTTFs above 20000 hours. Therefore, the baseline cluster architecture is optimal under the defined architectural constraints when reasonably reliable components are used.

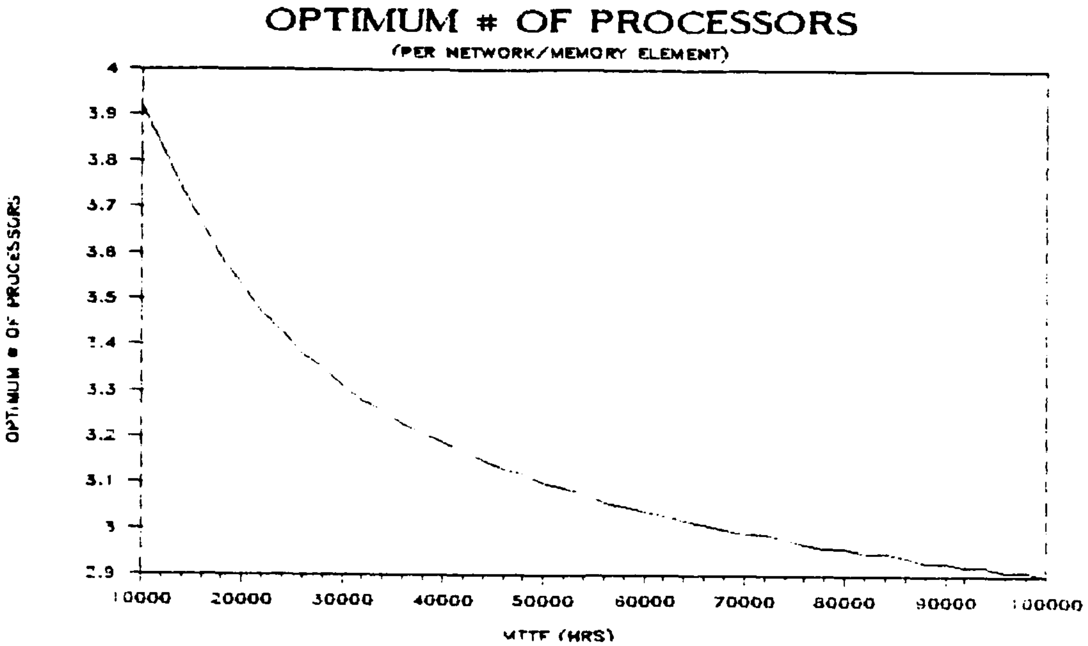


Figure 3.2-3: Optimum Number of Processors per Single Network/ Memory Element vs. Component MTTF.

3.2.3 Cluster Task Assignment

Assigning fewer tasks per cluster increases cluster reliability, but at a cost which can be paid in two ways by the total system. First: in order to make up for the decrease in tasks per cluster, more clusters can be added to the system. In this case, the probability of a cluster isolation increases since more clusters are being introduced in the system. Second: the same number of clusters may remain in the system. In this case, the total system taskload and, therefore, throughput decreases.

In a parallel processor, throughput and reliability are interchangeable quantities. This trade off will be examined in more detail in the next section (3.2.4).

Using the FTPP system model constructed for the fully linked baseline system (section 3.1), cluster reliabilities for two, three, four, and five task clusters were calculated for the assumptions of perfect and imperfect processor coverage. Results for both lower and upper bound assumptions are depicted graphically in figures 3.2-4 and 3.2-5 respectively. As expected, cluster reliability increases as fewer tasks are assigned to the cluster. The systems effect of redistributing and reducing the number of tasks per cluster by using additional clusters is examined in section 3.3.2.2.

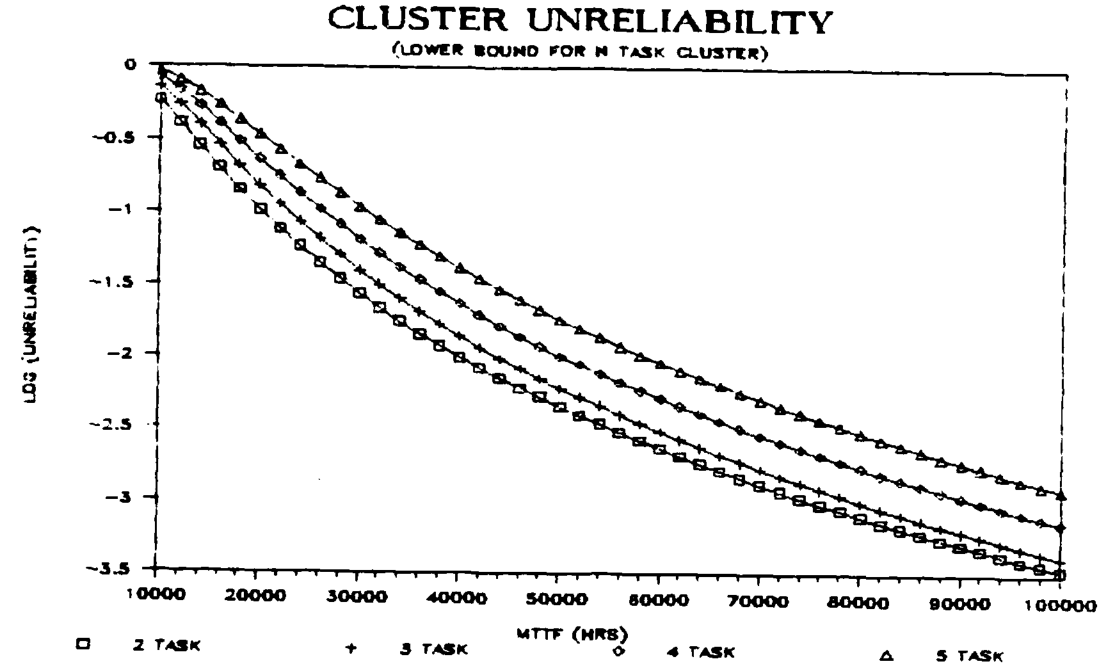


Figure 3.2-4: Cluster Unreliability vs. MTTF for an N Task Cluster (Lower Bound Assumptions).

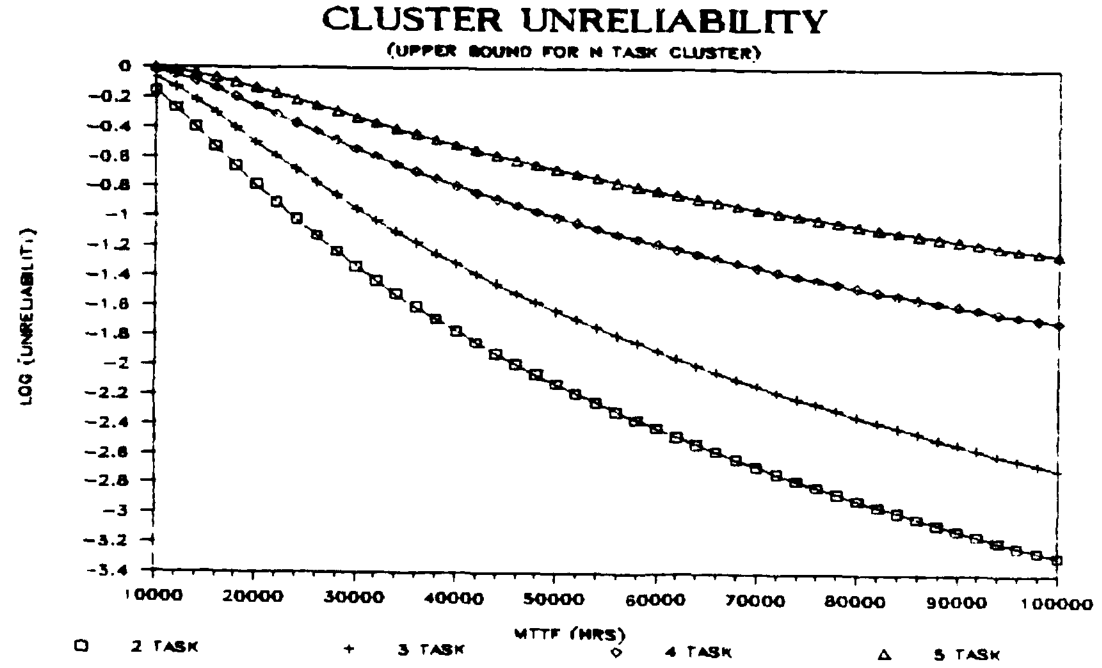


Figure 3.2-5: Cluster Unreliability vs. MTTF for an N Task Cluster (Upper Bound Assumptions).

3.2.4 Reliability/Throughput Trade Off

Reliability may be traded for throughput in a parallel system. The relationship may be fixed during the design process, or the trade off may occur during operation. As an example of the later: during critical computations, the FTPP may switch to a lower throughput mode to achieve a higher probability of success and switch back to the higher throughput mode when the critical computations are completed.

For the purposes of this thesis, the FTPP always works one problem. The term 'task' refers to the number of parallel computational paths that the problem is broken up into. Each parallel computational path may be performed by n processor elements. Figure 3.2-6 graphically depicts the expected speedup versus the number of tasks operating in parallel for the ideal case, with the lower and upper speedup bounds as defined by Hwang and Briggs (described in section 1.1). The differences are rather large. While 15 parallel tasks would ideally generate a speedup 15 times as fast, the lower and upper speedup bounds are only about 4 and 5 times as fast. The apparent jump in the graph for the n/ln(n) case at 2 tasks is an anomaly of the equation which is clearly impossible and should be ignored. Achieving a given speedup factor requires an extremely large number of parallel tasks. On the other hand, the loss of a task would represents a much smaller loss in throughput. The loss of a single task in an n task parallel system would ideally represent a throughput loss (T_{Loss}) of 1 times the throughput of a single processor (T_{sp}). Using the speedup

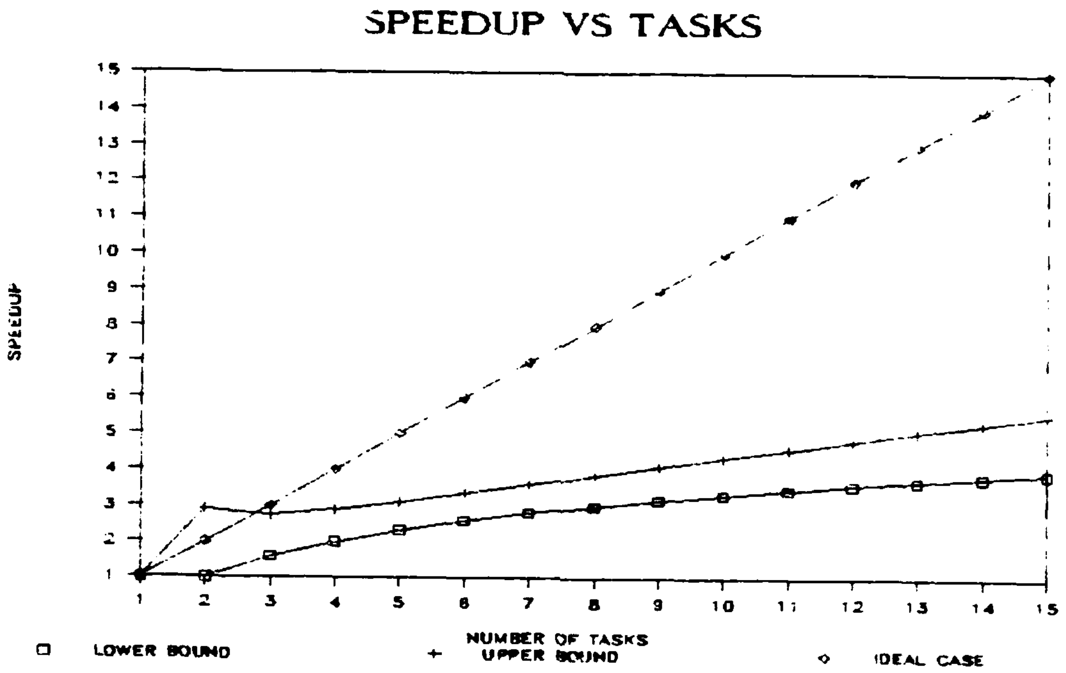


Figure 3.2-6: System Speedup vs. Number of Parallel Tasks.

estimates in section 1.1, the loss of a single task in a n cluster system would represent a throughput loss defined by the following equations:

$T_{Loss} = [\log_2(n/(n-1))]T_{sp}$ (Minsky's Conjecture) (3.2-6)

OR

$T_{Loss} = [(n/\ln(n))-((n-1)/\ln(n-1))]T_{sp}$ (Upper Bound) (3.2-7)

For a 15 task system, the expected throughput loss would be $(1)T_{sp}$ for the ideal case and between $(.1)T_{sp}$ and $(.23)T_{sp}$ using equations 3.2-6 and 3.2-7 respectively. Depending on the actual application requirement, the loss of at least one or more computational tasks should be tolerable.

In order to examine the relation between reliability and throughput, consider the set of 15 processors in the baseline cluster. The 15 processors can perform a minimur of 1 and a maximum of 15 tasks in parallel. If the set of processors perform 1 task, the FTPP problem will be solved using 1 computational path, with all 15 processors dedicated to the one path. If the set of processors perform 15 tasks, the FTPP problem will be solved using 15 computational paths, with each path performed by a single processor. Figure 3.2-7 graphically depicts processor unreliability as a function of the speedup (equivalent to the number of tasks in the ideal case) using the upper and lower bound processor assumptions described in section 3.1.2. As expected, an increase in

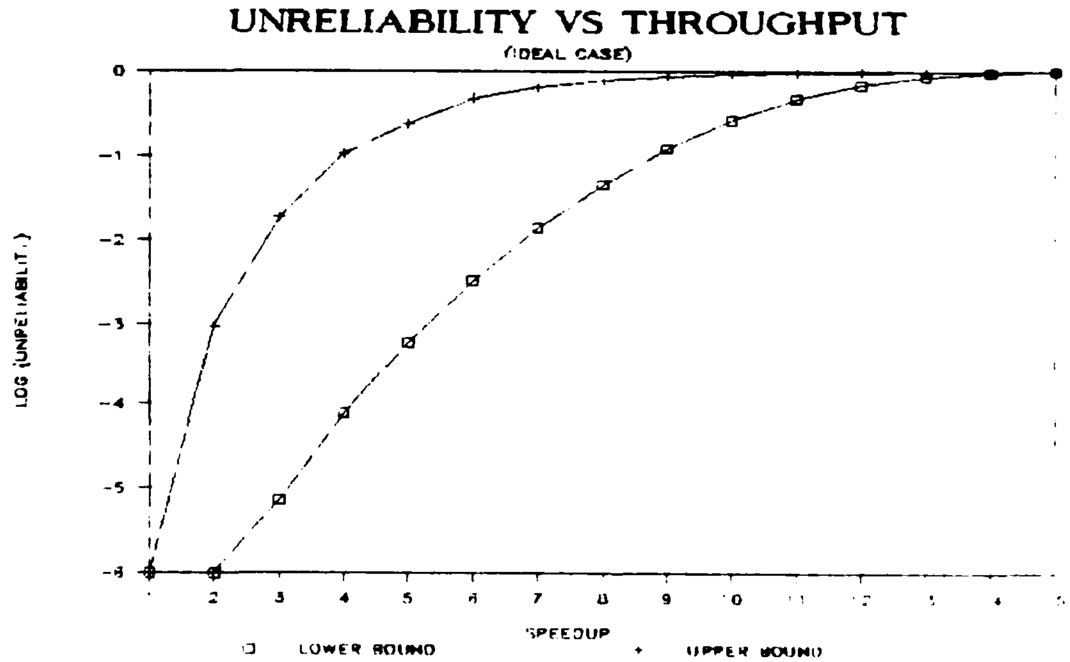


Figure 3.2-7: Unreliabilty vs. Throughput for a Set of 15 Processor Elements (Ideal Speedup Behavior).

throughput is accompanied by a decrease in reliability. Using upper bound processor assumptions, the cost of additional throughput in reliability loss is greater than for the lower bound case. If the Hwang and Briggs speedup bounds are used, the cost increases even more and both curves would rise more sharply.

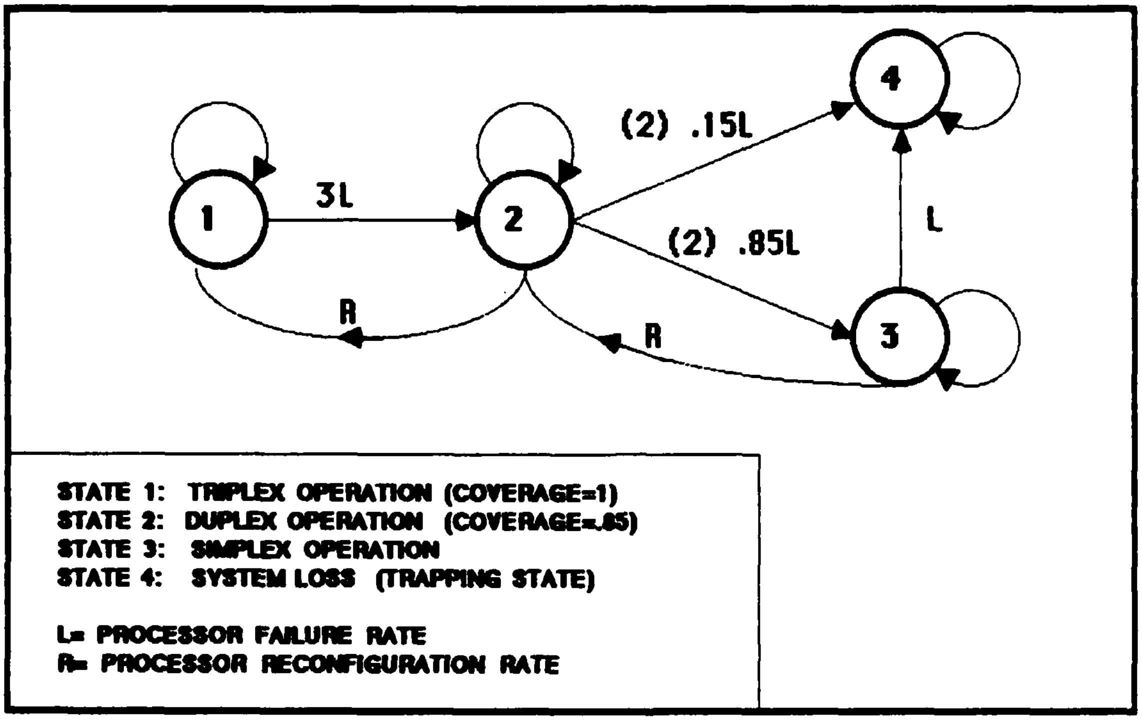
Power consumption is yet another term which may be added into the reliability/throughput trade off. Processors may be turned off when power consumption is critical at the cost of either reliability or throughput. The act of turning components on and off in an operational system may be risky in itself. Current operational spacecraft turn components on and off only when absolutely necessary.

The system throughput, reliability and power consumption may be 'tuned' by the designer of a parallel processor to achieve a mix which best meets the application requirement. The tuning is done by defining and redefining the delegation of processors to the required tasks.

3.2.5 Reliability Bottlenecks

The previous section showed that the loss of a computational task in a reasonably large parallel system is a tolerable event. This is not true for the tasks of cluster and global controller. The loss of a cluster controller results in cluster loss, and the loss of the global controller results in system loss. A cluster can never be more reliable than its cluster controller, and the system can never be more reliable than its global controller. The tasks of cluster and global controller can accurately be described as 'reliability bottlenecks' which deserve special attention. In the baseline architecture, three processor elements were reserved for each cluster controller, and four processor elements were reserved for the global controller. In view of the relative importance of tasks, it would be logical to transfer processor elements performing computational tasks to the tasks of cluster or global controller should a CC or GC experience degradation. Using this redundancy strategy effectively means the controllers have access to unlimited spares. Figure 3.2-8 diagrams the Markov Model of the case of a cluster controller with unlimited spares. While this strategy is certainly more reliable than a simple triplex of processor elements, the Markov Model shows that there is still the possibility of controller loss due to the finite reconfiguration time of the processors. If a second failure occurs before the processors can reconfigure, the controller may be lost.

Other possible reliability bottlenecks are the loss of a memory or network element causing the loss of a task. As discussed earlier, the dispersion of tasks between different memory and network elements can eliminate single point failures provided that the cluster is not overloaded with tasks.



STATE 1: TRIPLEX OPERATION (COVERAGE=1)
STATE 2: DUPLEX OPERATION (COVERAGE=.85)
STATE 3: SIMPLEX OPERATION
STATE 4: SYSTEM LOSS (TRAPPING STATE)

L= PROCESSOR FAILURE RATE
R= PROCESSOR RECONFIGURATION RATE

Figure 3.2-8: Markov Model Diagram of Cluster Controller with Unlimited Spares.

3.3 INTER-CLUSTER MODIFICATION OPTIONS

This section will address modifications at the system level and their effects upon system performance. Section 3.3.1 examines the relative performance of three selected cluster topologies. Section 3.3.2 examines cluster redundancy strategies and their effect on system reliability.

3.3.1 Cluster Topologies

Cluster topology refers to the particular method employed to link clusters. Cluster topology affects throughput, reliability maintainability, and modularity. Interconnection strategy is a key factor in obtaining high performance, in reducing cost, and in keeping the system feasible in terms of engineering [7].

Graph theory provides useful terminology in the analysis of cluster topologies. Clusters are defined as nodes and the distance between any pair of adjacent clusters is defined as 1. The diameter of the graph (k) is defined as the maximum of the length of the shortest paths between any two nodes. A graph of diameter k will take no more than k hops to travel between any two nodes. The fan-out (d) is defined as the number of connections emanating from each node providing all nodes have the same fan-out. One of the topologies examined in this thesis has a fan-out which varies beveen clusters. A graph having n nodes, diameter k, and fan-out d is defined as an (n,k,d) graph. Each graph can be redefined by examining t link failures. Link failures may increase diameter [8,9].

The diameter of a FTPP system is a measure of the speed of the network. As the diameter increases, so does the communications time between certain nodes. An increase in communication time decreases system throughput. The fan-out of a cluster is a measure of the complexity of a cluster. As the fan-out increases, so does the complexity of the nodes. An increase in nodal complexity translates to a decrease in nodal reliability though not necessarily system reliability. The fan-out is also a measure of the maintainability and modularity of a system. As the fan-out increases, it becomes increasingly complicated to replace failed clusters and to add new clusters.

For a given number of nodes, the most desirable graph would have a diameter and fan-out of one. This is only possible for the case of n=2. As n increases, the fan-out must increase to keep the diameter small or the diameter must increase to keep the fan-out small. The following section will analyze three cluster topologies in an attempt to work out the various trade offs involved in selecting an interconnection strategy. The three topologies to be examined include: centrally linked (star), fully linked (fully cross-strapped), and singly linked (ring) - (figure 3.3-1).

The reliability analysis in each of the following sections will calculate the probability of at least one cluster isolation P(ISOLATION) for the different topologies looking exclusively at input/output elements.

This probability can be used to estimate the FTPP system reliability (P(S)) using the results of section 3.1. In section 3.1, P(S) was derived for a fully linked system. P(ISOLATION), looking exclusively at input/output elements, was approximated by neglecting failures due to input/output elements of other clusters and found to be relatively accurate. Using these probabilities, the following equation holds true:

$$P(S) = (1-P(ISOLATION))xRC^{n} \qquad (3.3\text{-}1)$$

where:

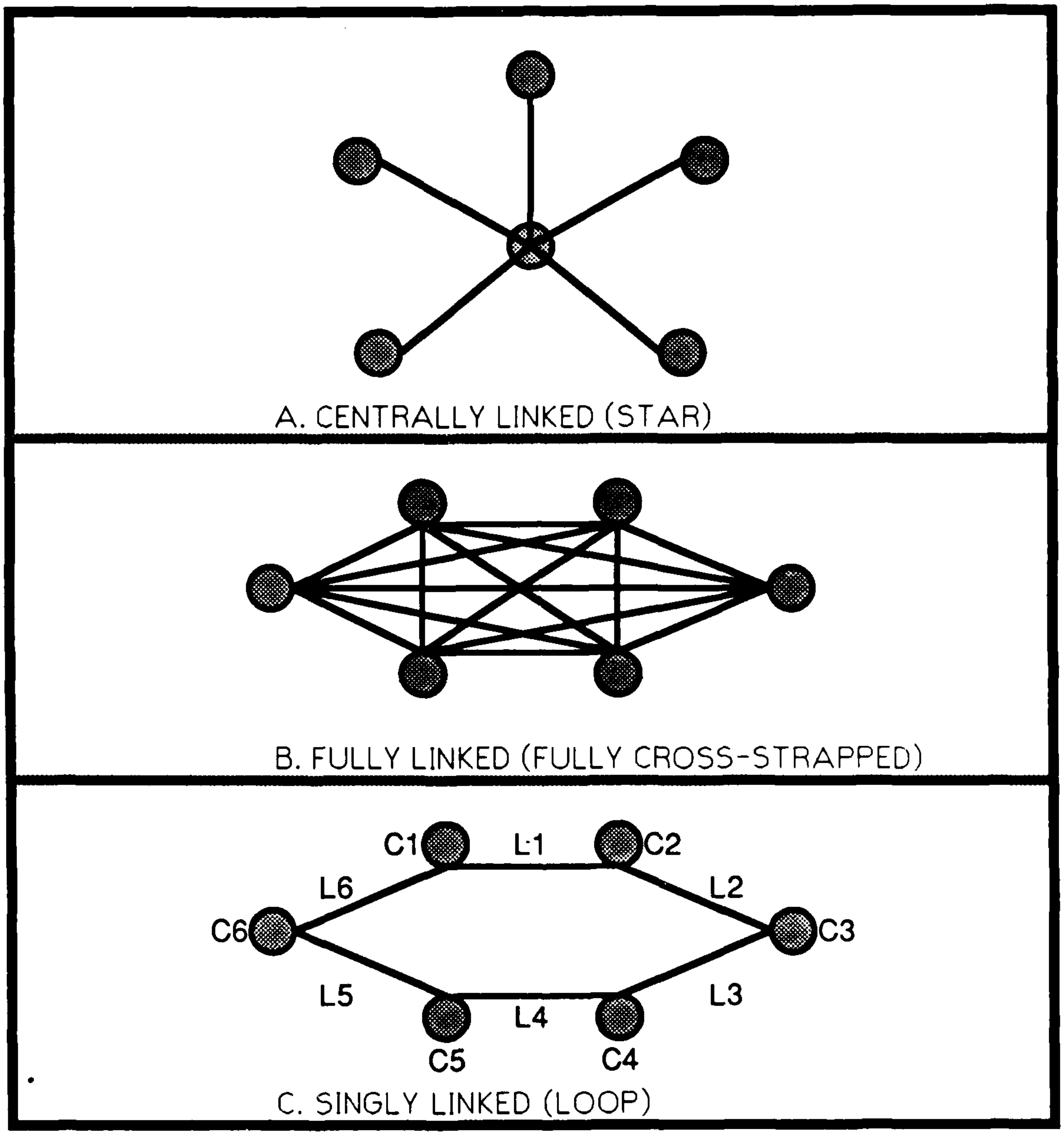
RC = The reliability of a cluster neglecting the effects of the input/output elements.

n = Number of clusters in the FTPP system.

Therefore, RC can be estimated as:

$$RC = [P(S)/(1-P(ISOLATION))]^{1/n} \qquad (3.3\text{-}2)$$

RC will remain constant for different cluster topologies since topology only affects input/output elements which are, by definition, independent of RC. Knowing P(ISOLATION) for each topology, equation 3.3-2 can be used to estimate RC and equation 3.3.1 used to calculate the total system reliability. The following analysis will address the baseline case of n=6 specifically and then generalize for all n.



A. CENTRALLY LINKED (STAR)

B. FULLY LINKED (FULLY CROSS-STRAPPED)

C. SINGLY LINKED (LOOP)

Figure 3.3-1: Cluster Topologies.

3.3.1.1 Centrally Linked Topology

Figure 3.3-1(A) depicts the centrally linked topology of 6 clusters. With no link failures (t=0): k=2 ,d=5 (for the central cluster), and d=1 (for the distributed clusters). The advantages of this topology are the low diameter and the low fan-out of the distributed clusters. The diameter remains two and the fan-out of the distributed clusters remains one no matter how many clusters are added to the network. The disadvantages of this topology are the low tolerance to link failures and the high fan-out of the central cluster. Any one link failure will cause an isolation, and the fan-out of the central cluster increases with the number of clusters [d=n-1]. While distributed clusters can be added and replaced relatively easily, the central cluster is lacking in maintainability.

The reliability of the centrally linked topology is relatively easy to calculate by decomposing on the input/output elements of the central cluster. Since any one link failure will cause a cluster isolation, the probability of at least one isolation is the complement of the probability of no link failures:

$$P(\text{ISOLATION}) = 1-[P(\text{0 LINK FAILURES})] \qquad (3.3\text{-}3)$$

P(0 LINK FAILURES) can be found by decomposing on the input/output elements of the central cluster.

$$P(\text{0 LINKS FAIL}) = \sum_{N=0}^{5} [P(\text{0 LINKS FAIL/N IO WORK}) \times P(\text{N IO WORK})] \quad (3.3\text{-}4)$$

where:

P(N IO WORK) = Probability that exactly N out of 5 input/output elements of the central cluster operate.

P(O LINKS FAIL/N IO WORK) = Probability that no links fail given that exactly N input/output elements of the central cluster operate.

P(N IO WORK) can be calculated combinatorially using equation 3.1-5 and substituting RIOC for RIO where RIOC equals the reliability of a single input/output element of the central cluster. The probability that no links fail, given that exactly N input/output elements of the central cluster operate, can be calculated by using the fact that every distributed cluster must possess at least 2 operational lines to the central cluster:

$$\text{P(O LINKS FAIL/N IO WORK)} = [\text{P(at least 2 of N IOD WORK)}]^{5} \quad (3.3\text{-}5)$$

where:

IOD = Input/output element of a distributed cluster.

P(at least 2 of N IOD WORK) is found using equation 3.1-14 and substituting RIOD and N for RIO and X respectively where RIOD equals the reliability of a single input/output element of a distributed cluster.

To generalize for n, the exponent in equation 3.3-5 becomes (n-1).

3.3.1.2 Fully Linked Topology

Figure 3.3-1(B) depicts the fully linked topology of 6 clusters which was modelled in section 3.1. With no link failures (t=0): k=1 and d=5 for all clusters. For t=1 to t=4: k=2. Five link failures may cause isolation depending on which links fail. The advantages of this topology are the low diameter and the high tolerance to link failures. The diameter remains one, and the tolerance to link failures increases as more clusters are added to the network. A fully linked network can tolerate at least n-2 link failures without an isolation. The major disadvantage of this network is the high fan-out of the clusters. The fan-out increases as more cluster are added to the system [k=n-1]. The network is clearly lacking in the areas of maintainability and modularity.

While the complexity associated with a fully linked network makes an exact reliability analysis difficult, the reliability can be bounded relatively tightly as was shown in section 3.1.1. A lower unreliability bound was generated by neglecting cluster isolations due to input/output failures in other clusters and multiplying the reliabilities of all clusters. An upper reliability bound was generated by taking into account all possible failure modes for each cluster and multiplying the reliabilities of all clusters. The dominant failure mode for a particular cluster isolation is the failure of that cluster's input/output elements and is why the reliability can be tightly bounded. This failure mode becomes increasingly dominant as n increases.

3.3.1.3 Singly Linked Topology

Figure 3.3-1(C) depicts the singly linked topology of 6 clusters. With no link failures (t=0): k=3 and d=2 for all clusters. The major advantage of this topology is the low fan-out of the clusters. The fan-out remains two no matter how many clusters are added to the network. The disadvantages of this topology are the low tolerance to link failures and the relatively high diameter. Any two link failures will always cause an isolation, and the diameter increases as more clusters are added to the network [k=n/2 (n even); k=(n-1)/2 (n odd)]. The network possesses both high maintainability and modularity.

The reliability of the singly linked network is the most difficult to analyze. Unlike the centrally linked architecture, the probability of an isolation can not be calculated using a simple decomposition. Unlike the fully linked network, the isolation of a particular cluster is not dominated by failures of its own input/output elements. The fact that adjacent link failures are dependent must be taken into account. The probability of an isolation can be calculated using combinatorial methods and conditional probability.

Since any two link failures will cause an isolation, the probability of at least one isolation, P(ISOLATION), is the complement of the probability of zero or one link failure:

$$P(\text{ISOLATON}) = 1-[P(\text{0 LINKS FAIL}) + P(\text{1 LINK FAILS})] \qquad (3.3\text{-}6)$$

To calculate the probability of zero link failures, conditional probability must be used. Link failures are dependent and each link's probability of operating is conditional on the state of the other links:

$$P(0\ LINKS\ FAIL) = P(L1) x P(L2/L1) x P(L3/L1 \cap L2) x P(L4/L1 \cap L2 \cap L3) x$$

$$P(L5/L1 \cap L2 \cap L3 \cap L4) x P(L6/L1 \cap L2 \cap L3 \cap L4 \cap L5) \qquad (3.3\text{-}7)$$

where:

$P(LX/LA \cap \ldots \cap LE)$ = Probability that link X operates given that links A through E operate.

The probability link 1 operates is the probability that at least 2 lines in the link operate. For a line to operate, the input/output elements on both ends must function.

$$P(L1) = 1-[\ (1-(RIO)^2)^5 + 5(1-(RIO)^2)^4(RIO)^2\ ] \qquad (3.3\text{-}8)$$

where: RIO = Probability that a single input/output element operates.

The probability that link 2 operates, given link 1 operates, can be calculated using conditional probability:

$$P(L2/L1) = P(L1 \cap L2) \ /\ P(L1) \qquad (3.3\text{-}9)$$

P(L1) has been calculated in equation 3.3-8. The probability that two adjacent links operate, $P(L1 \cap L2)$, can be calculated by decomposing on the input/output elements of the cluster the two links share (cluster 2):

$$P(L1 \cap L2) = \sum_{N=0}^{5} [P(L1 \cap L2/N\ IO\ WORK) \times P(N\ IO\ WORK)] \qquad (3.3\text{-}10)$$

where:

$P(L1 \cap L2/N\ IO\ WORK)$ = Probability that L1 and L2 operate given that exactly N input/output elements of the shared cluster operate.

P(N IO WORK) has been calculated in equation 3.1-5. P(L1 ∩ L2/N IO WORK) can be calculated using the fact that at least two operational lines must operate between clusters 1 and 2 and between clusters 2 and 3:

$$P(L1 \cap L2/N\ IO\ WORK) = P(\text{at least 2 of } N\ IO \text{ of cluster 1 WORK}) \times P(\text{at least 2 of } N\ IO \text{ of cluster 3 WORK}) \quad (3.3\text{-}11)$$

The probability that at least 2 of N input/output elements of a cluster operate can be calculated using equation 3.1-14 and substituting N for X.

The probability that link 3 operates, given that links 1 and 2 operate, is equivalent to the probability that link 3 operates given that link 2 operates. The state of link 1 has no effect on link 3 because they are independent. Only adjacent links are dependent. Therfore:

$$P(L3/L1 \cap L2) = P(L3/L2) = P(L2/L1) \quad (3.3\text{-}12)$$

Using the same line of reasoning:

$$P(L2/L1) = P(L3/L1 \cap L2) = P(L4/L1 \cap L2 \cap L3) = P(L5/L1 \cap L2 \cap L3 \cap L4) \quad (3.3\text{-}13)$$

These equalities all describe the probability that a link operates given that one adjacent link operates. The probability that link 6 operates, given that links 1 to 5 operate, describes the probability a link operates given that both adjacent links operate. Once again, conditional probability is employed:

$$P(L6/L1 \cap L2 \cap L3 \cap L4 \cap L5) = P(L6/L5 \cap L1) = P(L1 \cap L5 \cap L6) / P(L1 \cap L5) \quad (3.3\text{-}14)$$

Since links 1 and 5 are non-adjacent, hence independent, the probability they both operate is the product of the reliabilities:

$$P(L1 \cap L5) = P(L1) x P(L5) = P(L1)^2 \qquad (3.3\text{-}15)$$

P(L1) has been calculated in equation 3.3-8. The probability that links 1 and 5 and 6 operate can be calculated by decomposing on the input/output elements of one cluster shared by 2 links, and then decomposing on the input/output elements of the other cluster shared by two links. In this case, the first decomposition is done on the elements of cluster 6, and the second decomposition is done on the elements of cluster 1.

Decomposing on the elements of cluster 6 yields the equation:

$$P(L1 \cap L5 \cap L6)$$
$$= \sum_{N=0}^{8} [P(L1 \cap L5 \cap L6/N \text{ IO of C6 WORK}) \times P(N \text{ IO of C6 WORK})] \qquad (3.3\text{-}16)$$

P(N IO WORK) has been calculated in equation 3.1-5. The probability that links 1 and 5 and 6 operate, given that N IO elements of cluster 6 operates, can be calculated using the fact that at least two lines of each of the three links must operate.

$$P(L1 \cap L5 \cap L6/(0 \text{ or } 1) \text{ IO of C6 WORK}) = 0 \qquad (3.3\text{-}17)$$

To calculate the remaining elements, a second decomposition is done on the elements of cluster 1:

$$P(L1 \cap L5 \cap L6/2 \text{ IO of C6 WORK}) =$$
$$\sum_{N=1}^{8} [P(L1 \cap L5 \cap L6/2 \text{ IO of C6 and N IO of C1 WORK}) x P(N \text{ IO of C1 WORK})] \qquad (3.3\text{-}18)$$

P(N IO WORK) has been calculated in equation 3.1-5. P(L1 ∩ L5 ∩ L6/X IO of C6 WORK and Y IO of C1 WORK) is calculated by determining the product of the probabilities of three conditions: 1. The probability that a sufficient number of input/output elements of cluster 5 operate to achieve a working link between clusters 5 and 6. 2. The probability that a sufficient number of input/output elements of cluster 2 operate to achieve a working link between clusters 2 and 1. 3. The probability that the X operating elements in cluster 6, and Y operating elements in cluster 1, will yield a working link between clusters 6 and 1:

P(L1 ∩ L5 ∩ L6/2 IO of C6 WORK and N IO of C1 WORK) =

P(at least 2 of 2 IO of C5 WORK)xP(at least 2 of N IO of C2 WORK)x

P(LINK is achieved between C6 and C1) (3.3-19)

P(at least 2 of N IO WORK) can be calculated using equation 3.1-14

and substituting N for X:

P(Link is achieved between C6 and C1) = 0,0,.1,.3,.6,1 for

N = 0,1,2,3,4,5 respectively.

Using the same line of reasoning, the remaining elements can be calculated:

P(L1 ∩ L5 ∩ L6/3 IO of C6 WORK) =

$$\sum_{N=1}^{5} [P(L1 \cap L5 \cap L6/3 \text{ IO of C6 and N IO of C1 WORK}) x P(\text{N IO of C1 WORK})] \quad (3.3\text{-}20)$$

where:

P(L1 ∩ L5 ∩ L6/3 IO of C6 WORK and N IO of C1 WORK) =

P(at least 2 of 3 IO of C5 WORK)xP(at least 2 of N IO of C2 WORK)x

P(LINK is achieved between C6 and C1) (3.3-21)

P(Link is achieved between C6 and C1) = 0,0,.3,.7,1,1 for

N = 0,1,2,3,4,5 respectively.

$$P(L1 \cap L5 \cap L6/4\ \text{IO of C6 WORK}) =$$

$$\sum_{N=1}^{5} [P(L1 \cap L5 \cap L6/4\ \text{IO of C6 and N IO of C1 WORK}) \text{x} P(\text{N IO of C1 WORK})] \qquad (3.3\text{-}22)$$

where:

$$P(L1 \cap L5 \cap L6/4\ \text{IO of C6 WORK and N IO of C1 WORK}) =$$

$$P(\text{at least 2 of 4 IO of C5 WORK}) \text{x} P(\text{at least 2 of N IO of C2 WORK}) \text{x}$$

$$P(\text{LINK is achieved between C6 and C1}) \qquad (3.3\text{-}23)$$

P(Link is achieved between C6 and C1) = 0,0,.6,1,1,1 for

N = 0,1,2,3,4,5 respectively.

$$P(L1 \cap L5 \cap L6/5\ \text{IO of C6 WORK}) =$$

$$\sum_{N=1}^{5} [P(L1 \cap L5 \cap L6/5\ \text{IO of C6 and N IO of C1 WORK}) \text{x} P(\text{N IO of C1 WORK})] \qquad (3.3\text{-}24)$$

where:

$$P(L1 \cap L5 \cap L6/5\ \text{IO of C6 WORK and N IO of C1 WORK}) =$$

$$P(\text{at least 2 of 5 IO of C5 WORK}) \text{x} P(\text{at least 2 of N IO of C2 WORK}) \text{x}$$

$$P(\text{LINK is achieved between C6 and C1}) \qquad (3.3\text{-}25)$$

P(Link is achieved between C6 and C1) = 0,0,1,i,1,1 for

N = 0,1,2,3,4,5 respectively.

The probability that exactly one link fails can be calculated by arbitrarily examining the case where only link 2 fails. As was done previously:

$$P(\overline{L2}\ \text{ONLY}) = P(L1) \text{x} P(\overline{L2}/L1) \text{x} P(L3/L1 \cap \overline{L2}) \text{x} P(L4/L1 \cap \overline{L2} \cap L3) \text{x}$$

$$P(L5/L1 \cap \overline{L2} \cap L3 \cap L4) \text{x} P(L6/L1 \cap \overline{L2} \cap L3 \cap L4 \cap L5) \qquad (3.3\text{-}26)$$

Since any one link of six links may fail with equal probability, and these events are mutually exclusive:

$$P(1\ \text{LINK FAILS}) = 6(P(\overline{L2}\ \text{ONLY})) \qquad (3.3\text{-}27)$$

Using the reasoning employed previously:

$$P(L4/L1 \cap \overline{L2} \cap L3) = P(L4/L3) \qquad (3.3\text{-}28)$$

$$P(L5/L1 \cap \overline{L2} \cap L3 \cap L4) = P(L5/L4) \qquad (3.3\text{-}29)$$

$$P(L6/L1 \cap \overline{L2} \cap L3 \cap L4 \cap L5) = P(L6/L1 \cap L5) \qquad (3.3\text{-}30)$$

$$P(L2/L1) = P(L4/L3) = P(L5/L4) \qquad (3.3\text{-}31)$$

$P(L1)$, $P(L2/L1)$, and $P(L5/L1 \cap L5)$ have been calculated previously

$P(\overline{L2}/L1)$ is simply the complement of $P(L2/L1)$:

$$P(\overline{L2}/L1) = 1-[P(L2/L1)] \qquad (3.3\text{-}32)$$

The only remaining element to calculate is $P(L3/L1 \cap \overline{L2})$. Since links 1 and 3 are independent, the calculation becomes the probability that a link operates given that an adjacent link fails. As was done previously, conditional probability is employed:

$$P(L3/\overline{L2}) = P(L3 \cap \overline{L2}) \ / \ P(\overline{L2}) \qquad (3.3\text{-}33)$$

$P(\overline{L2})$ is the complement of $P(L1)$ which has been calculated previously.

$$P(\overline{L2}) = 1-[P(L1)] \qquad (3.3\text{-}34)$$

The probability link 3 operates and link 2 fails can be found by decomposing on the input/output elements of the shared cluster (cluster 3):

$$P(L3 \cap \overline{L2}) = \sum_{N=0}^{8} [P(L3 \cap \overline{L2}/N\ \text{IO WORK}) \times P(N\ \text{IO WORK})] \qquad (3.3\text{-}35)$$

where:

$$P(L3 \cap \overline{L2}/N \text{ IO WORK}) = P(\text{at least } N\text{-}1 \text{ of } N \text{ IO of C2 FAIL}) \times P(2 \text{ of } N \text{ IO of C4 WORK}) \qquad (3.3\text{-}36)$$

P(N IO WORK) and P(at least 2 of N IO WORK) have already been calculated in equations 3.1-5 and 3.1-14. P(at least N-1 of N IO FAIL) can be calculated combinatorially:

$$P(\text{at least } N\text{-}1 \text{ of } N \text{ IO FAIL}) = \sum_{i=0}^{1} \left[\binom{N}{i} (1-RIO)^{N-i} (RIO)^{i} \right] \qquad (3.3\text{-}37)$$

To generalize for n, the probabilities that 0 and 1 links fail become:

$$P(0 \text{ LINKS FAIL}) = P(L) \times P(L/L)^{n-2} \times P(L/L \cap L) \qquad (3.3\text{-}38)$$

$$P(1 \text{ LINK FAILS}) = N \times P(L) \times P(\overline{L}/L) \times P(L/\overline{L}) \times P(L/L)^{n-4} \times P(L/L \cap L) \qquad (3.3\text{-}39)$$

3.3.1.4 Topology Comparisons

This section attempts to quantify the performance of the three topologies in the areas of throughput, maintainability, modularity, and reliability. Comparisons will be made for the baseline case of six clusters specifically and for the general case of n clusters. For the purposes of this comparison, all clusters are assumed to communicate with all other clusters with equal probability.

In the area of throughput, the fully linked system is clearly the most desirable. Any two clusters can communicate directly (1 hop). The centrally linked system requires an average of 1.67 and the singly linked system an average of 1.80 hops to communicate between two clusters. As n increases, the relative rankings remain unchanged, and the differences become increasingly pronounced. For the fully linked system, k=1 always. For the centrally linked system:

$$k_{avg} = [(n-1) + 2(\binom{n}{2} - (n-1))]/ \binom{n}{2} \qquad (3.3\text{-}40)$$

which simplifies to:

$$k_{avg} = 2 - [2/(n-1)] + [2/n(n-1)] = 2 \ (\text{as } n \to \infty) \qquad (3.3\text{-}41)$$

For the singly linked system, k_{avg} increases without bound as $n \to \infty$.

In the areas of maintainability and modularity, the fully linked system is clearly the least desirable. Replacing a cluster requires 25 disconnections and connections. Adding a cluster requires 25 connections. The centrally linked system requires an average of 8.35 and the singly linked an average of 10 disconnections and connections to replace a cluster. Adding a cluster requires only 5 connections for the centrally linked system while the singly linked system requires 10 disconnections and 10 connections. As n increases, the relative rankings remain the same: 5(n-1) disconnections and connections are required to replace a cluster and 5(n-1) connections are required to add a cluster for the fully linked system; an average of 5(2n-2)/n disconnections and reconnections are required to replace a cluster and 5 connections are required to add a cluster for the centrally linked system; and 10 disconnections and reconnections are required to replace or add a cluster for the singly linked system.

In the area of reliability, the relationship between the complexity of the input/output elements and their failure rates is a determining factor in the relative reliabilities of the topologies. The reliability models for the three topologies derived in the previous sections were programmed in FORTRAN. Comparisons were made for the baseline case of 6 clusters. The system unreliabilities using both lower and upper bound assumptions are depicted graphically in figures 3.3-2 and 3.3-3 respectively. Using lower bound assumptions, the unreliability curves for the singly and fully linked systems are close, but the singly linked system becomes noticeably more reliable for MTTFs greater than 60000 hours. Using upper bound assumptions, the unreliability curves of the singly and fully linked systems are nearly identical but the raw data shows the singly linked system slightly more reliable. In both cases, the centrally linked system is least reliable.

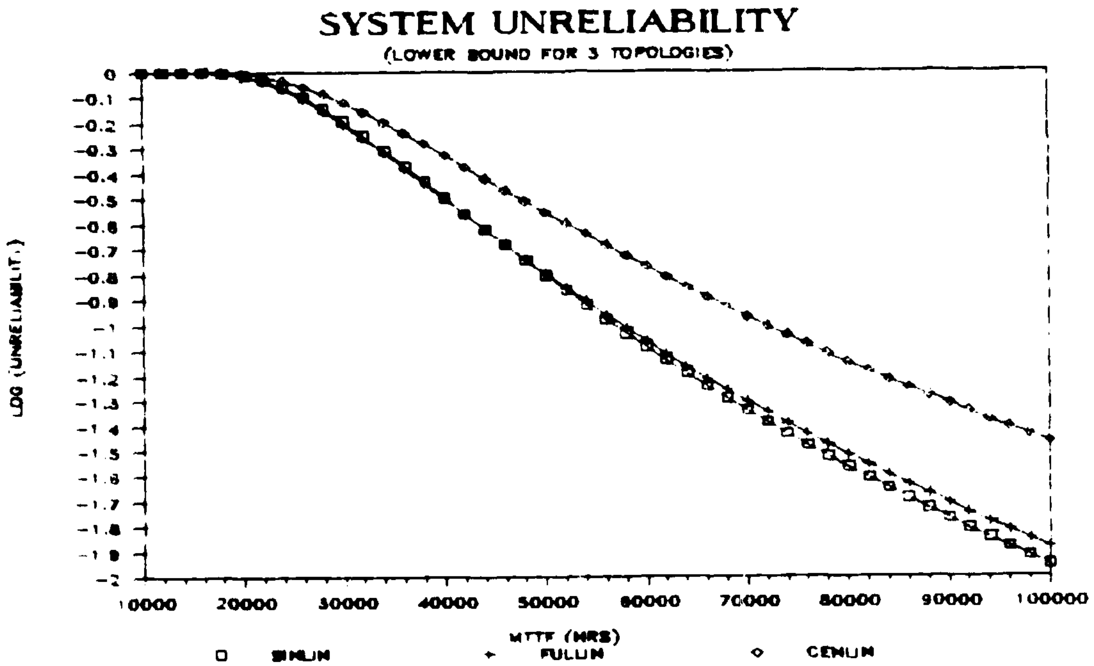


Figure 3.3-2: System Unreliability for 3 Topologies (6 Clusters and Lower Bound Assumptions).

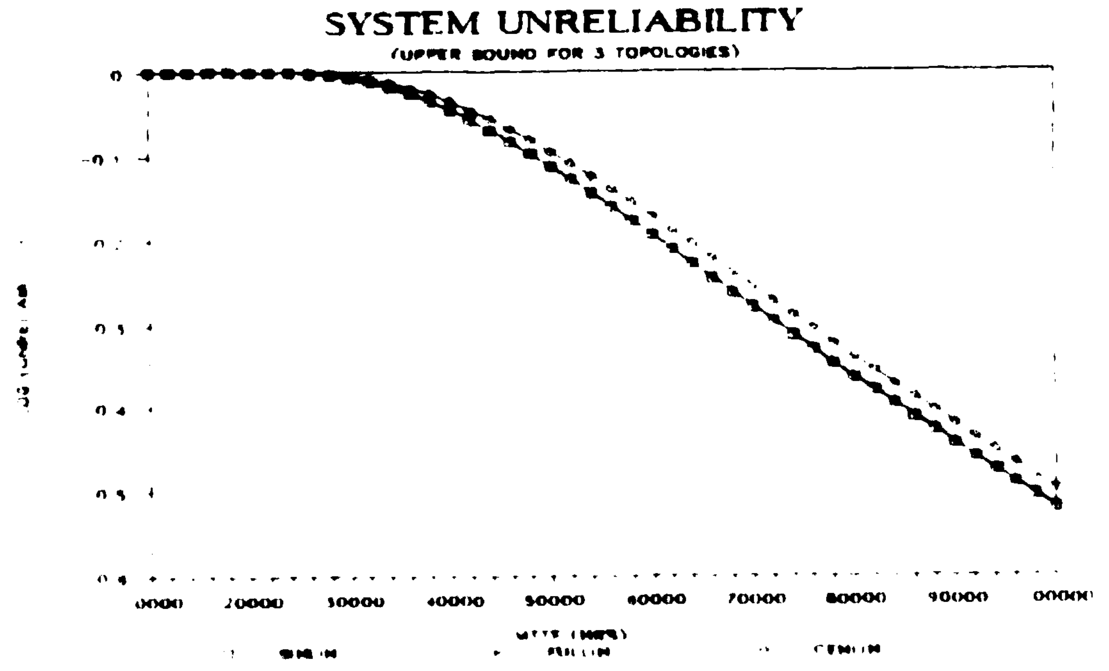


Figure 3.3-3: System Unreliability for 3 Topologies (6 Clusters and Upper Bound Assumptions).

For architectures with more clusters, the singly linked system would be expected to perform increasingly better relative to the other topologies since the complexity of the input/output elements remains constant. Figures 3.3-4 and 3.3-5 depict the unreliabilities of the three topologies as the number of clusters increases using an MTTF of 60000 hours for lower and upper assumptions respectively. The singly linked system is clearly superior in both cases. The fully and centrally linked systems exhibit a crossover at n=16 and n=15 clusters in the lower and upper bound graphs respectively. Before the crossover, the fully linked is more reliable than the centrally linked; after the crossover, the opposite is true. To examine the effects of a change in MTTF, figures 3.3-6 and 3.3-7 depict the unreliabilities of the three topologies using an MTTF of 100000 hours for lower and upper bound assumptions respectively. The graphs exhibit the same relative characteristics as the MTTF=60000 hours case, but the crossover point increases to n=20 and n=19 clusters in the lower and upper bound graphs respectively. In all cases, as n increases, the singly and centrally linked systems tend to parallel each other while the singly and fully linked systems tend to diverge.

The fully linked baseline case modelled in section 3.1 was undesireable in terms of the application reliability requirement. Modifying the topology to a singly linked system does not affect the reliability of the FTPP significantly enough to make a difference in terms of this requirement.

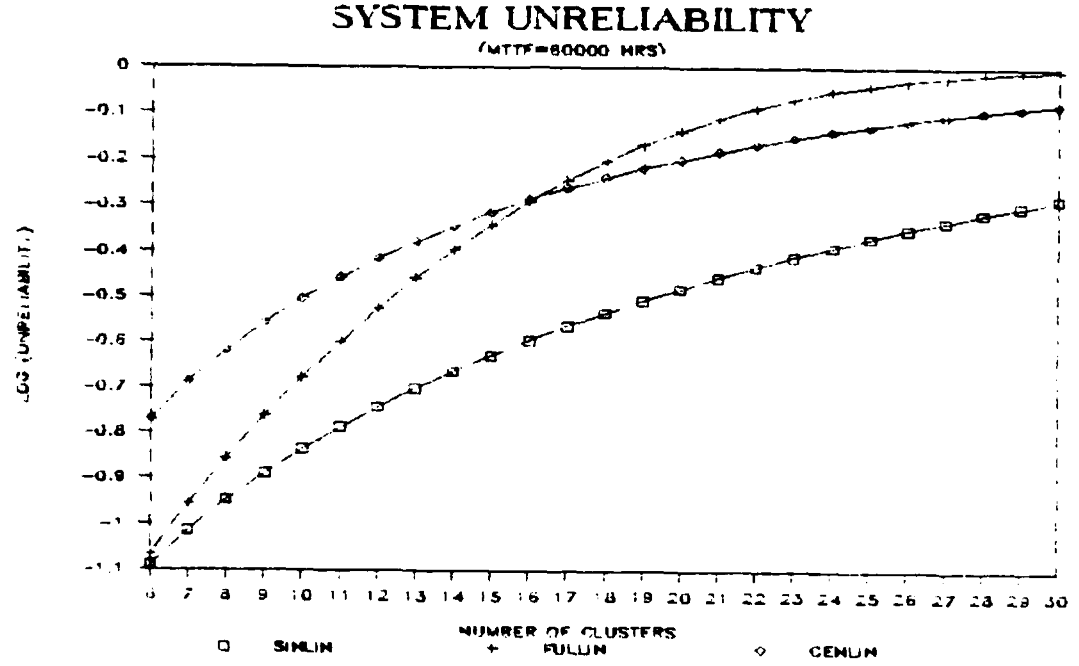


Figure 3.3-4: System Unreliability for 3 Topologies and N Clusters (MTTF=60000 hours and Lower Bound Assumptions).

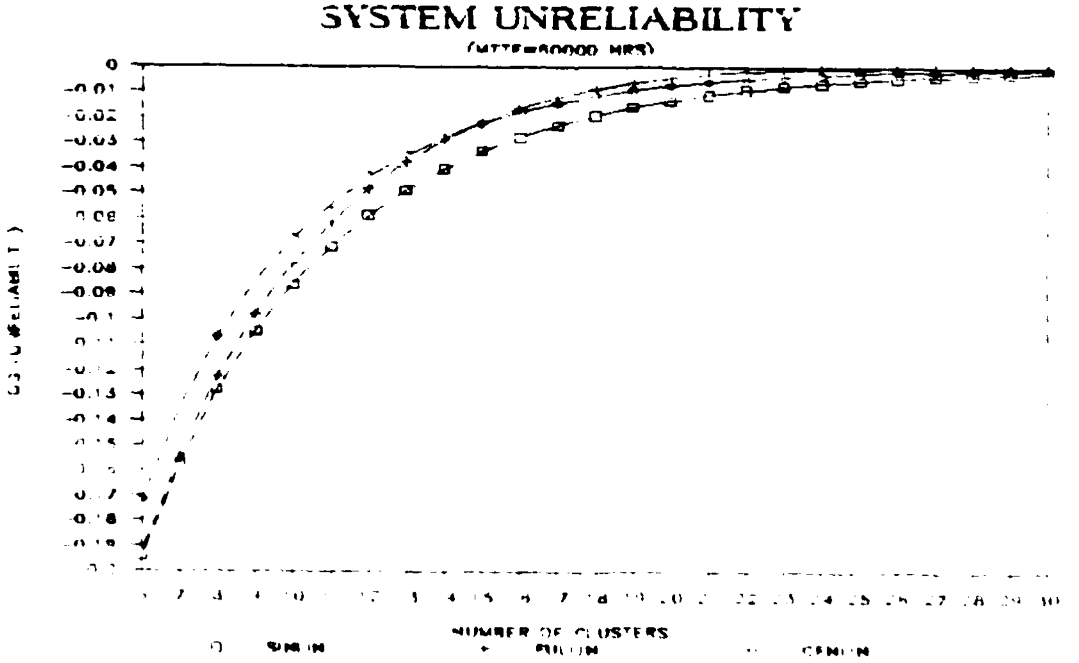


Figure 3.3-5: System Unreliability for 3 Topologies and N Clusters (MTTF=60000 hours and Upper Bound Assumptions).

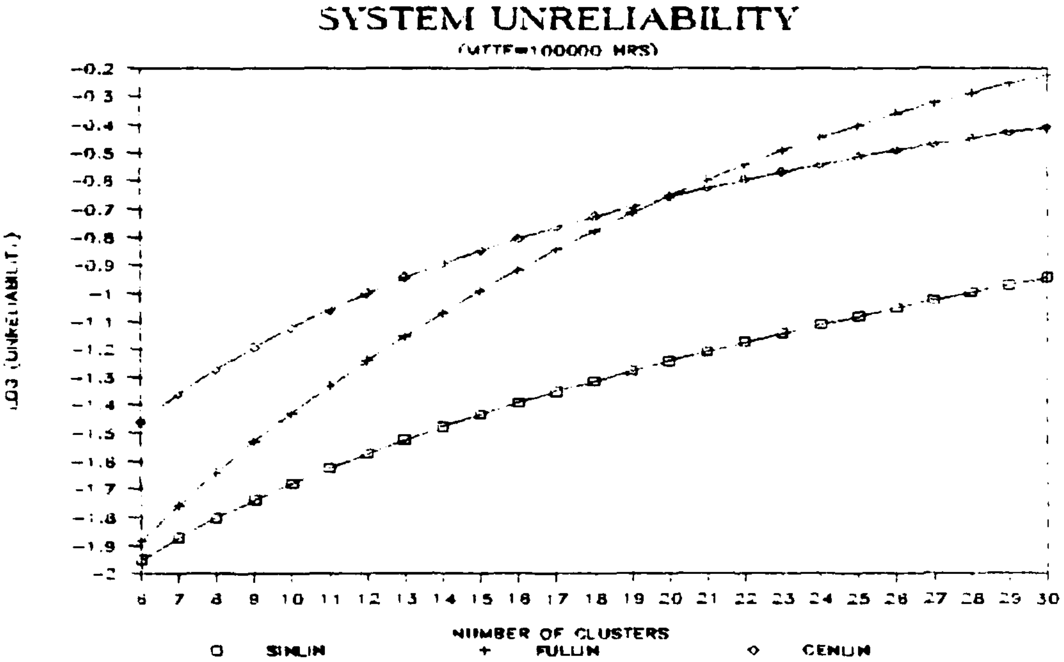


Figure 3.3-6: System Unreliability for 3 Topologies and N Clusters (MTTF=100000 hours and Lower Bound Assumptions).

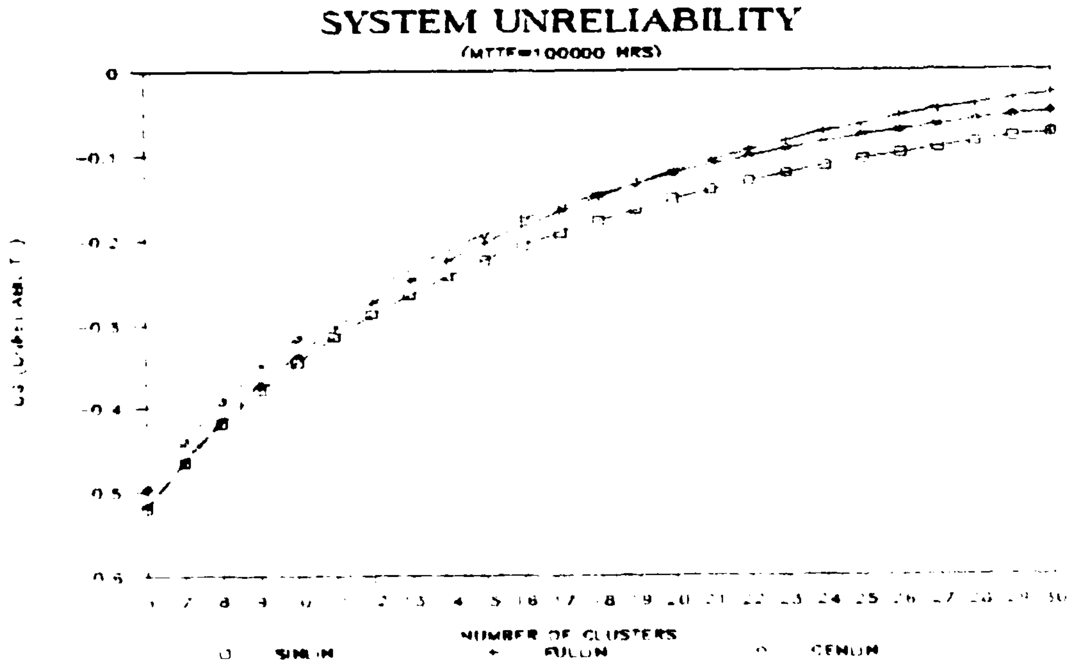


Figure 3.3-7: System Unreliability for 3 Topologies and N Clusters (MTTF=100000 hours and Upper Bound Assumptions).

3.3.2 Cluster Redundancy Strategies

Section 3.3.1 compared various cluster topologies under the assumption that any cluster failure or isolation constituted system failure. This section examines the case where some cluster failures/isolations are tolerated. A cluster failure/isolation may be acceptable in a system where there are redundant clusters or where degradation of system throughput is acceptable.

3.3.2.1 Toleration of Cluster Failure/Isolation

The methods and equations of section 3.3.1 will be employed to derive the reliability of a FTPP system of n clusters where 1 cluster failure or isolation is tolerated. The unreliability of the system (Q(S)) now becomes:

Q(S) = [P(0 ISOLATIONS) x P(2 OR MORE CLUSTERS FAIL)]+
[P(1 ISOLATION)x(P(2 OR MORE CLUSTERS FAIL)+((n-1)/n)P(1 CLUSTER FAILS))]+
[P(2 OR MORE ISOLATIONS)] (3.3-42)

The ((n-1)/n) factor is used to take into account the case where the isolated cluster is also the failed cluster. Most of the terms of equation 3.3-42 have already been derived.

P(0 ISOLATIONS) = 1-P(ISOLATION) (3.3-43)

P(ISOLATION) is the probability of at least one cluster isolation and has been derived for the various topologies in section 3.3-1:

P(2 OR MORE CLUSTERS FAIL) = 1 - [RC^n + nRC^{n-1}(1-RC)] (3.3-44)

$$P(1\ \text{CLUSTER FAILS}) = nRC^{n-1}(1-RC) \qquad (3.3\text{-}45)$$

RC is the reliability of a cluster, neglecting the effects of the input/output elements, and is defined in equation 3.3-2. The two remaining terms are related:

$$P(2\ \text{OR MORE ISOLATIONS}) = P(\text{ISOLATION}) - P(1\ \text{ISOLATION}) \qquad (3.3\text{-}46)$$

The remainder of this section is devoted to calculating the probability of exactly one isolation P(1 ISOLATION) for the various topologies.

For the centrally linked system, to find P(1 ISOLATION), the probability that one of the five distributed clusters becomes isolated is calculated. Since this may occur in 5 different ways, the resultant probability is multiplied by 5. Decomposing on the input/output elements of the central cluster yields the equation:

$$P(1\ \text{ISOLATION}) = 5\sum_{N=0}^{5}[P(1\ \text{ISOLATION/N IO WORK})\text{x}P(\text{N IO WORK})] \qquad (3.3\text{-}47)$$

P(N IO WORK) is the probability that exactly N out of 5 input/output elements of the central cluster operate, and can be calculated combinatorially using equation (3.1-5). The probability of exactly one isolation, given that N input/output elements of the central cluster operate, is the probability that there are at least two operational lines to four of the distributed clusters and less than two operational lines to one of the distributed clusters.

$$P(1\ \text{ISOLATION/N IO WORK}) = P(\text{at least 2 of N IO WORK})^{4}\ \text{x}$$
$$P(\text{less than 2 of N IO WORK}) \qquad (3.3\text{-}48)$$

P(at least 2 of N IO WORK) is found using equation 3.1.14 and substituting

RIOD and N for RIO and X respectively. P(less than 2 IO WORK) is simply the complement of P(at least 2 of N IO WORK). To generalize for n, the 5 preceding the summation in equation 3.3-47 becomes n-1 and the exponent on equation 3.3-48 becomes n-2.

For the fully linked system, clusters are assumed independent. The probability that an arbitrary cluster is isolated, P(1 CLUSTER ISOLATION), has already been defined by a lower and upper bound in section 3.1.1. The probability that one of six clusters becomes isolated becomes:

$$P(1\ \mathrm{ISOLATION}) = 6[1-P(1\ \mathrm{CLUSTER\ ISOLATION})]^5 \times [P(1\ \mathrm{CLUSTER\ ISOLATION})] \qquad (3.3\text{-}49)$$

P(1 ISOLATION) will also be defined by a lower and upper bound depending on the values of P(1 CLUSTER ISOLATION) used in equation 3.3-49. To generalize for n, the 6 becomes an n and the exponent becomes n-1 in equation 3.3-49.

For the singly linked system, allowing a cluster isolation means that any two adjacent link failures are tolerated. To find P(1 ISOLATION), the probability that two adjacent links fail is calculated. Since this may occur in 6 different ways, the resultant probability is multiplied by 6. Arbitrarily selecting links 2 and 3 as failed links yields:

$$P(1\ \mathrm{ISOLATION}) = 6 \mathrm{x} P(L1) \mathrm{x} P(\overline{L2}/L1) \mathrm{x} P(\overline{L3}/L1 \cap \overline{L2}) \mathrm{x} P(L4/L1 \cap \overline{L2} \cap \overline{L3}) \mathrm{x}$$
$$P(L5/L1 \cap \overline{L2} \cap \overline{L3} \cap L4) \mathrm{x} P(L6/L1 \cap \overline{L2} \cap \overline{L3} \cap L4 \cap L5) \qquad (3.3\text{-}50)$$

All terms in equation 3.3-50 have been calculated in section 3.3.1.3 with the exception of $P(\overline{L3}/L1 \cap \overline{L2})$ which is the probability that a link fails, given that an adjacent link fails. Using conditional probability:

$$P(\overline{L3}/\overline{L2}) = P(\overline{L3} \cap \overline{L2}) \;/\; P(\overline{L2}) \qquad (3.3\text{-}51)$$

$P(\overline{L2})$ is the complement of P(L1) which has been calculated previously. The probability link 3 fails and link 2 fails can be found by decomposing on the input/output elements of the shared cluster (cluster 3).

$$P(\overline{L3} \cap \overline{L2}) = \sum_{N=0}^{5} [P(\overline{L3} \cap \overline{L2}/N \text{ IO WORK}) \times P(N \text{ IO WORK})] \qquad (3.3\text{-}52)$$

where:

$$P(\overline{L3} \cap \overline{L2}/N \text{ IO WORK}) = P(\text{at least } N\text{-}1 \text{ of } N \text{ IO of C2 FAIL}) \times$$

$$P(\text{at least } N\text{-}1 \text{ of } N \text{ IO of C4 FAIL}) \qquad (3.3\text{-}53)$$

P(N IO WORK) and P(at least N-1 of N IO FAIL) have already been calculated in equations 3.1-5 and 3.3-37. To generalize for n, equation 3.3-50 becomes:

$$P(1 \text{ ISOLATION}) = n \times P(L) \times P(\overline{L}/L) \times P(\overline{L}/\overline{L}) \times P(L/\overline{L}) \times P(L/L)^{n-5} \times$$

$$P(L5/L \cap L) \qquad (3.3\text{-}54)$$

The unreliabilities of the three topologies with a redundant cluster were programmed in FORTRAN. Comparisons were made for the baseline case of 6 clusters plus 1 redundant cluster (the preceding derivation examined 5 clusters plus 1 redundant cluster). The system unreliabilities using both lower and upper bound assumptions are depicted graphically in figures 3.3-8 and 3.3-9 respectively. Using lower bound assumptions, the fully linked system is most reliable followed by the singly and centrally linked systems respectively. Using upper bound assumptions, the singly and fully linked systems are nearly identical while the centrally linked is least reliable. The raw data shows the singly linked system slightly more reliable. While the assumptions of perfect and imperfect processor coverage should not affect the relative reliabilities of the topologies, the change can be accounted for by the fact that the probability of isolation for the fully linked system was calculated with the additional upper and lower bounds concerning input/output elements described in section 3.3.1.

With a single redundant cluster, using lower bound assumptions and component MTTFs under 10^5 hours (10^4 hours for network elements), the fully linked system is able to meet the duplex computer reliability requirement. The singly linked system is able to meet the triplex computer requirement and the centrally linked system is able to meet the simplex computer requirement. Using upper bound assumptions under the same conditions, all three topologies comes closer to, but do not meet, the quadruplex computer requirement.

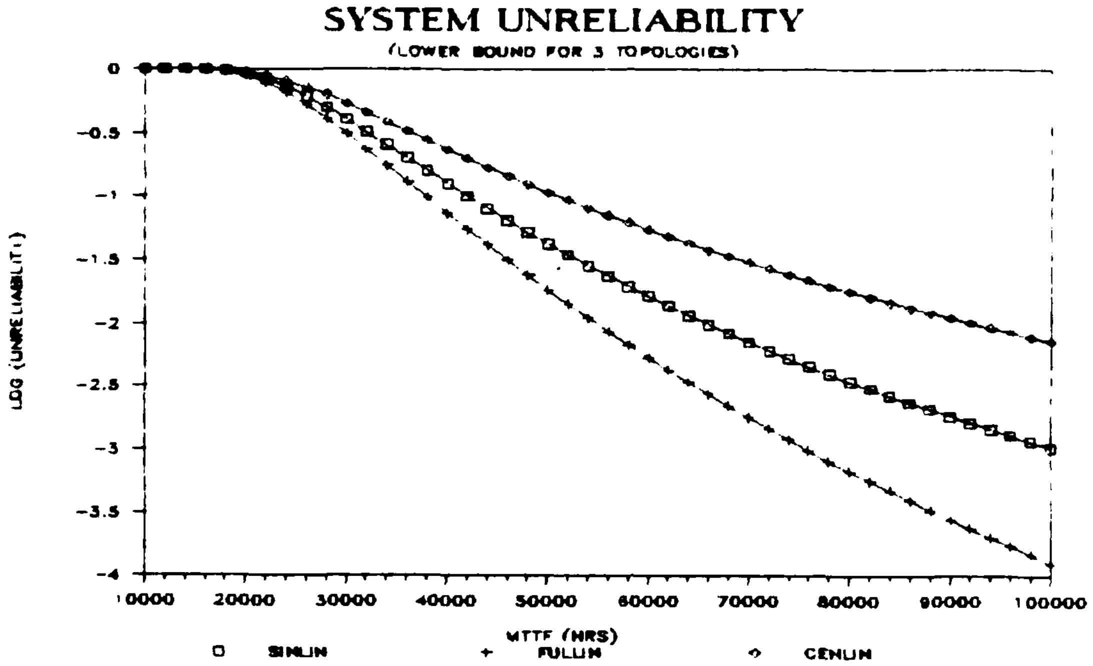


Figure 3.3-8: System Unreliability for 3 Topologies (6 Clusters plus 1 Spare and Lower Bound Assumptions).

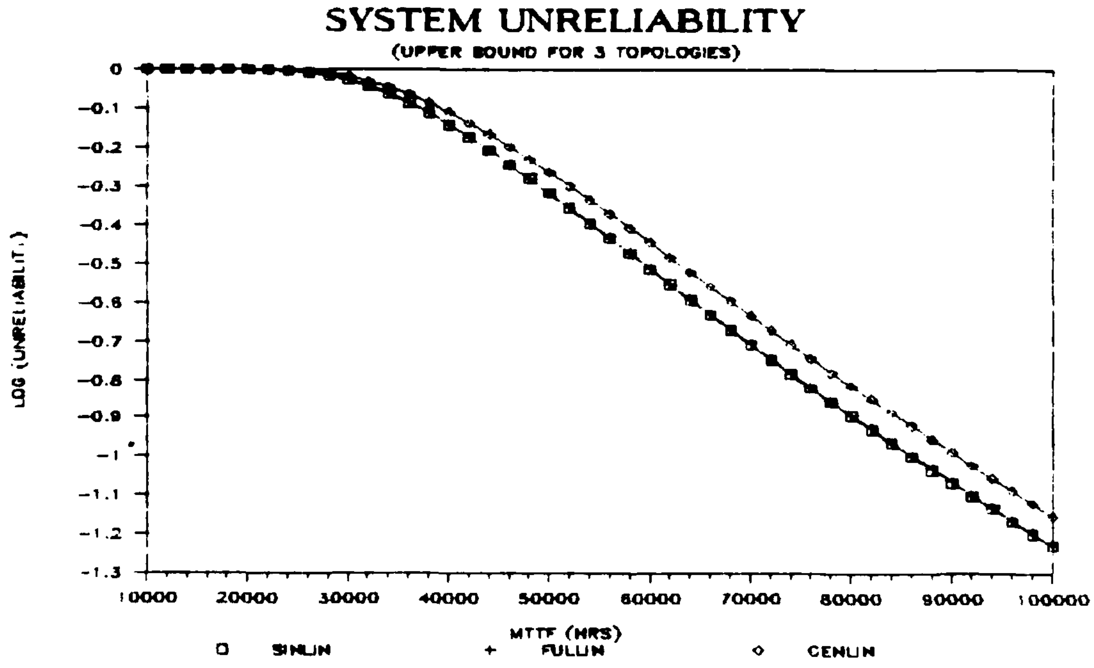


Figure 3.3-9: System Unreliability for 3 Topologies (6 Clusters plus 1 Spare and Upper Bound Assumptions).

3.3.2.2 Replication of Clusters

The previous section (3.3.2.1) examined the case where a single cluster failure/isolation was tolerated in a six cluster system. In terms of system reliability calculations, this is identical to the case of a five cluster system plus one redundant cluster. The methods employed in the previous section may be used to determine the reliability of a system with N redundant clusters. These methods produce exceedingly elaborate equations as the number of redundant clusters increases. As an example: for a six cluster system with two redundant clusters, the system's unreliability (QS) becomes:

QS = [P(0 ISOLATIONS)xP(3 OR MORE CLUSTERS FAIL)]+
[P(1 ISOLATION)x(P(3 OR MORE CLUSTERS FAIL)+(21/28)P(2 CLUSTERS FAIL))]+
[P(2 ISOLATIONS)x(P(3 OR MORE CLUSTERS FAIL)+(27/28)P(2 CLUSTERS FAIL)+
(6/8)P(1 CLUSTER FAILS))]+[P(3 OR MORE ISOLATIONS)] (3.3-55)

P(3 OR MORE ISOLATIONS) has not been calculated previously and will result in elaborate, though manageable, equations.

A relatively simple case to examine is the fully connected baseline system where the assumption of cluster independence permits the system reliability (RS) to be expressed as a function of the number of redundant clusters (N):

$$RS = \sum_{I=0}^{N} \binom{6+N}{I} RC^{6+N-I}(1-RC)^{I} \qquad (3.3\text{-}56)$$

where:

RC = Reliability of a single cluster.

Lower and upper bounds can be calculated by using the lower and upper bound calculations on cluster reliability respectively.

The system unreliability of the fully connected baseline system plus N redundant clusters, using both lower and upper bound assumptions, is depicted graphically in figures 3.3-10 and 3.3-11. These figures show that for the given values of N, the system reliability increases as more clusters are added to the system, but at a decreasing rate. As more clusters are added, the complexity and unreliability of the input/output elements increase. This effect becomes increasingly dominant as N increases and raises the question as to whether there is a limit to the number of redundant clusters that can be added to a fully connected system and still be able to increase system reliability.

Figures 3.3-12 and 3.3-13 depict the system unreliability as a function of the number of redundant clusters, with six clusters required to perform the assigned tasks, for the fully connected baseline system using both lower and upper bound assumptions respectively. The x-axis scale depicts the total number of clusters in the system. Curves were generated for MTTFs of 20000, 25000 and 30000 hours. The curves show there is indeed a limit to the number of redundant clusters which can increase system reliability. Using lower bound assumptions, the limit is approximately 10 redundant clusters for MTTF=20000 hours, 13 clusters for MTTF=25000 hours, and 16 clusters for MTTF=30000 hours. Using upper bound assumptions, the limit is approximately 12 redundant clusters for MTTF=20000 hours, 17 clusters for MTTF=25000 hours, and 20 clusters for MTTF=30000 hours. The limit increases as the MTTFs of the system components increase. The unreliability curves exhibit the effect of diminishing marginal returns. Each additional redundant cluster produces a smaller increase in reliability than the previous one (on a log scale).

Using lower bound assumptions and component MTTFs under 10^8 hours (10^6 hours for network elements), the fully linked system is able to meet the simplex computer reliability requirement with two redundant clusters. Using upper bound assumptions under the same conditions, the fully linked system is able to meet the triplex computer requirement with two redundant cluster and the duplex computer requirement with four redundant clusters.

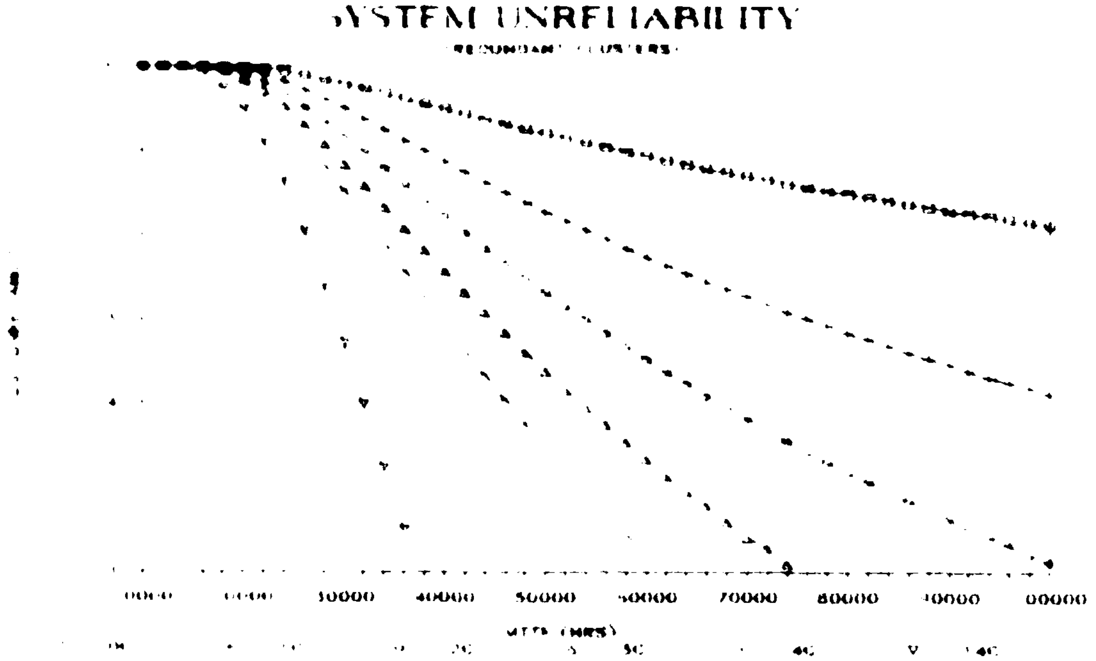


Figure 3.3-10: System Unreliability for Baseline System with Redundant Clusters (Lower Bound Assumptions).

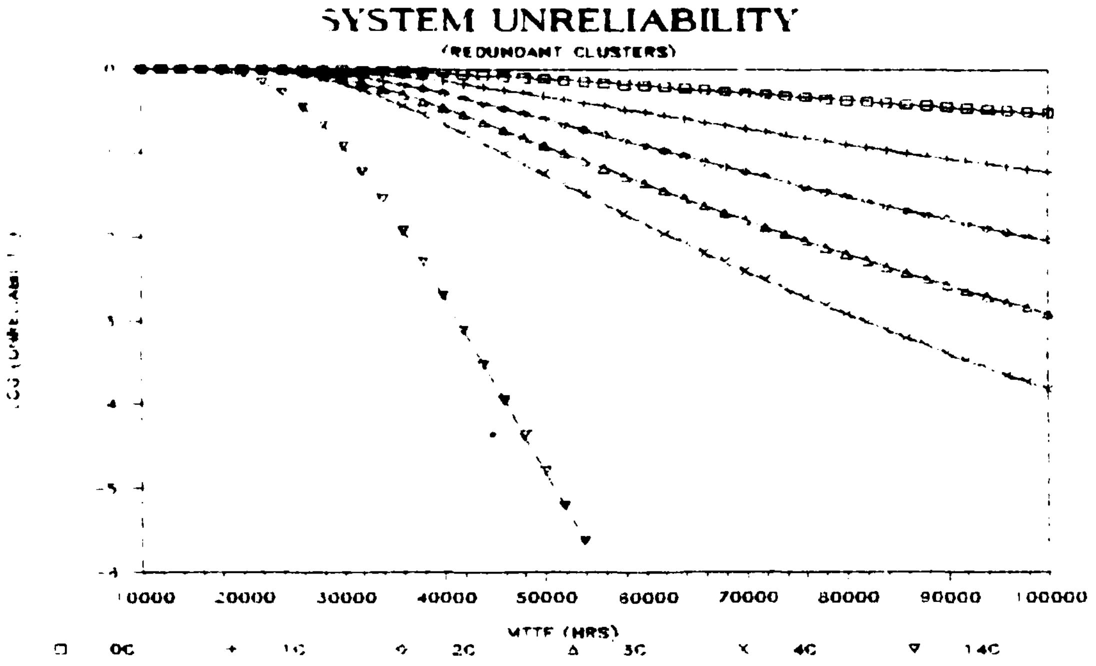


Figure 3.3-11: System Unreliability for Baseline System with Redundant Clusters (Upper Bound Assumptions).

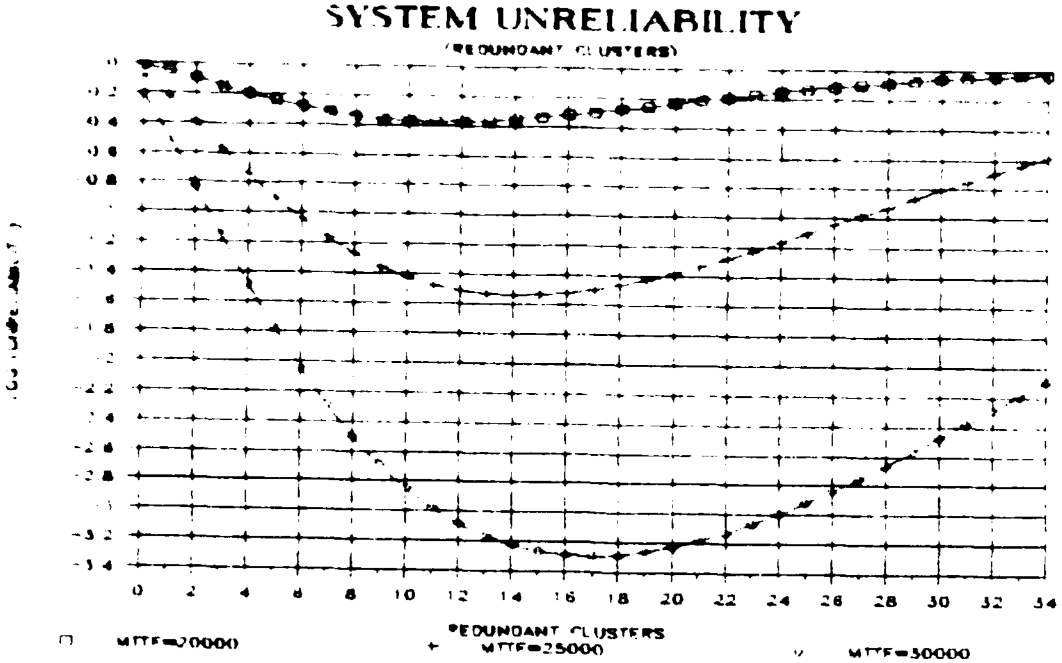


Figure 3.3-12: System Unreliability for Baseline System with N Redundant Clusters (Lower Bound Assumptions).

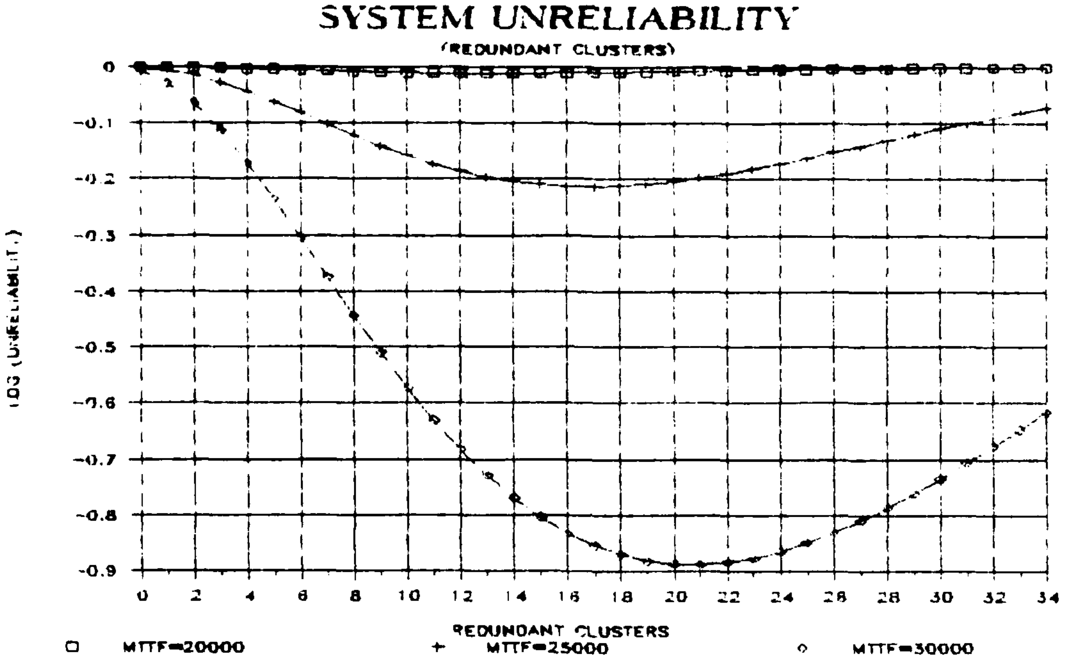


Figure 3.3-13: System Unreliability for Baseline System with N Redundant Clusters (Upper Bound Assumptions).

3.3.2.3 Distribution oi Tasks Among Added Clusters

Section 3.2.3 examined the change in cluster reliability as a result of assigning different numbers of tasks to the cluster. This section will examine the systems effect of decreasing the number of tasks per cluster and using additional clusters in order to keep the system throughput constant. As was discussed in section 3.2.3, using fewer tasks per cluster tends to increase cluster reliability. Adding more clusters to the system, however, tends to decrease system reliability. Four FTPP configurations will be examined using the FTPP system model constructed for the fully linked baseline system (section 3.1). The first configuration is the baseline case of six clusters, three of which are assigned four tasks and three of which are assigned five tasks. The second configuration consists of seven clusters, each of which is assigned four tasks. The third configuration consists of eleven clusters, ten of which are assigned three tasks and one of which is assigned two tasks. The fourth configuration consists of twenty-one clusters, each of which is assigned two tasks. In each case, the four configurations perform 20 computational tasks and are, therefore, capable of the same throughput. The difference between the four configurations is the way the tasks are distributed throughout the system.

The system reliability was calculated for the assumptions of perfect and imperfect processor coverages discussed in section 3.1.2. Results for both lower and upper bound assumptions are depicted graphically in figures

3.3-14 and 3.3-15 respectively. Using lower bound assumptions, the six cluster system is most reliable followed by the seven cluster, eleven cluster, and twenty-one cluster systems. Using upper bound assumptions, the eleven cluster system is most reliable followed by the seven cluster, six cluster and twenty-one cluster systems. In both cases, all curves tend to diverge from one another with the exception of the twenty-one cluster curve which tends to converge with all other curves.

Using lower bound assumptions and component MTTFs under 10^5 hours (10^4 hours for network elements), the distribution of tasks among added clusters decrease system reliability to the point where not even the quadruplex computer reliability requirement is met for the twenty-one cluster case. Using upper bound assumptions under the same conditions, only the eleven cluster system comes close to meeting the quadruplex computer requirement.

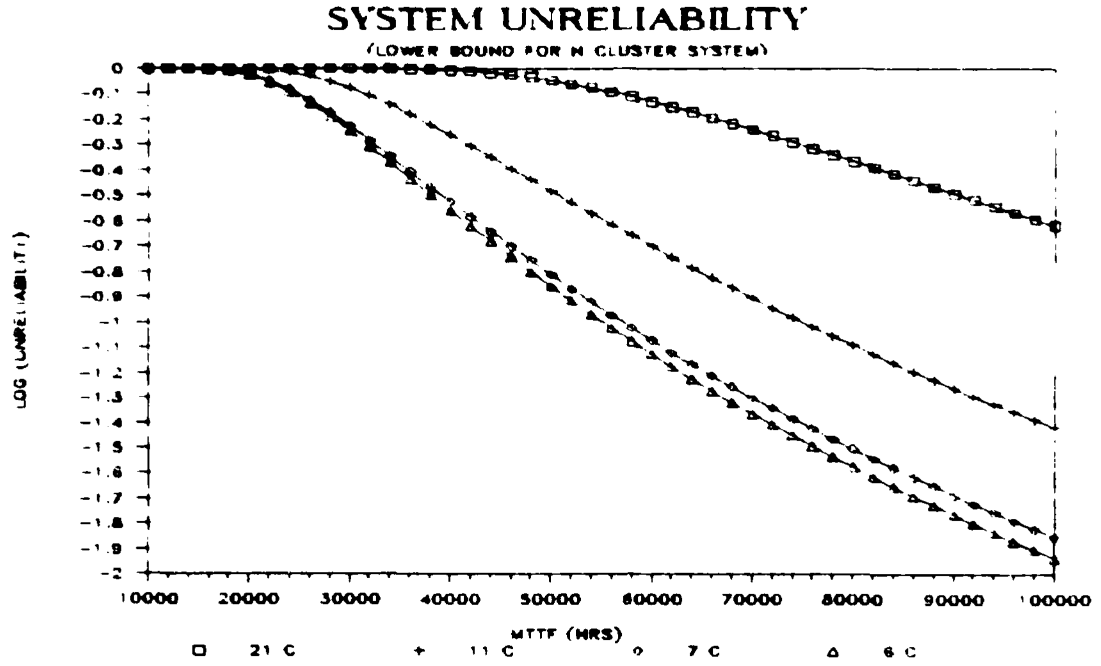


Figure 3.3-14: System Unreliability for 20 Task N Cluster System (Lower Bound Assumptions).

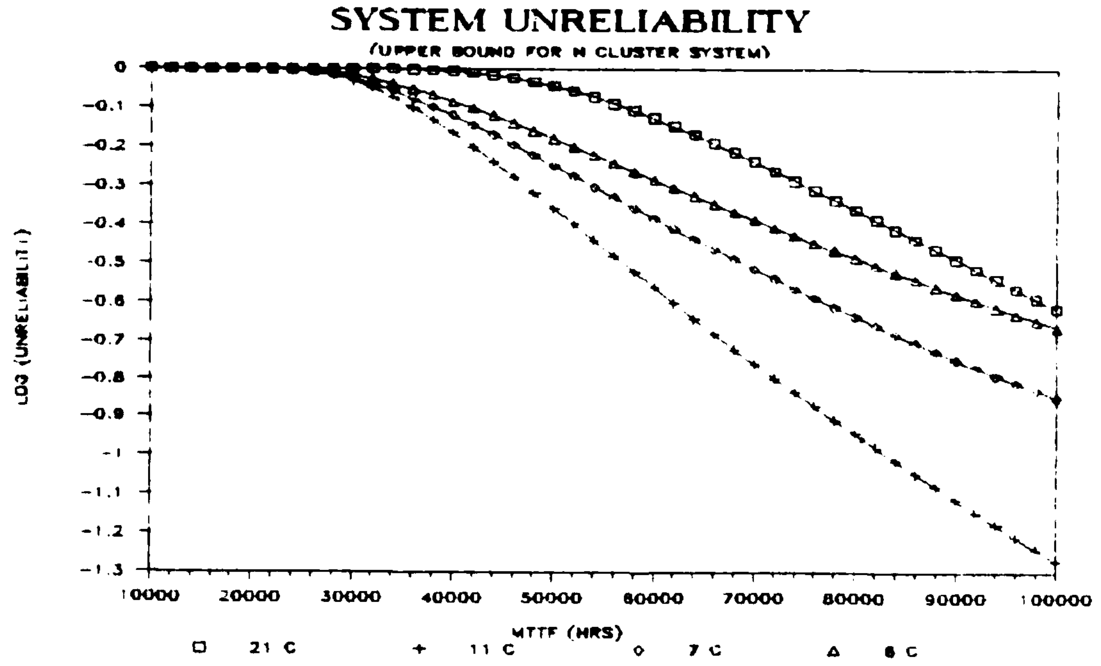


Figure 3.3-15: System Unreliability for 20 Task N Cluster System (Upper Bound Assumptions).

CHAPTER 4

CONCLUSIONS/RECOMMENDATIONS

4.1 CONCLUSIONS

This thesis has modelled a FTPP system architecture based upon a potential space system application requirement. The architecture utilizes four basic building block components: memory elements, processor elements, network elements, and input/output elements. The FTPP architectural and redundancy strategies were perturbed in attempts to define a more reliable system. These perturbations produced relationships between FTPP reliability, throughput, and architecture. A description of these relationships and the modelling techniques used to derive them should prove useful to a system-designer attempting to meet a specific application requirement.

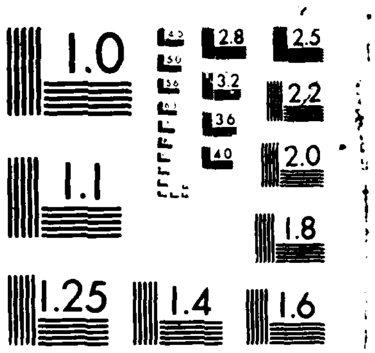
In good design practice, the choice of reliable components is essential [6]. This is particularly true for the FTPP processor and network elements, which provide the greatest delta increase in cluster reliability of the four FTPP building blocks. Since processor elements provide a greater delta increase when upper bound processor assumptions are used (upper bound assumptions will more likely predict actual processor behavior than will lower bound assumtions) and since network elements are the simplest components used in the FTPP and will probably be the most difficult element to improve upon in terms of failure rate, the use of the most reliable processor elements should be particularly stressed in the design phase. The possibility of improving network element reliability through duplication is being considered.

In defining a cluster architecture, the number of processor elements per network and memory must be chosen. There exists an optimum number of processors that will maximize cluster reliability, and as component failure rate decreases, the optimum number of processors will also decrease. For the range of parameters examined in this thesis, the optimal number of processors was four for component MTTFs below 20000 hours (200000 hours for network elements) and three for MTTFs above 20000 hours. Before defining an architecture, the designer should perform an optimization for the component parameters used. The optimization will depend on the component failure rates, processor coverage, and the degree that added component complexity affects component failure rate.

Both cluster controllers and global controllers represent reliability bottlenecks that deserve special attention. A cluster can not tolerate a cluster controller failure, and the system can not tolerate a global controller failure. In a system with realistic computational speedup assumptions, the loss of a set of processors assigned a computational task represents a relatively small decrease in system throughput which the system should tolerate (the number of computational tasks is equal to the number of parallel computational paths the FTPP job is partitioned into).

AD-A185 967 RELIABILITY MODELLING FOR FAULT-TOLERANT PARALLEL PROCESSOR(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH E I GJERMUNDSEN JAN 87 AFIT/CI/NR/87-51T 2/2

UNCLASSIFIED F/G 12/6 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

Therefore, processors performing computational tasks should be able to perform a controller task should a controller experience degradation. Also, to prevent single point failures, tasks should be dispersed to the fullest extent possible among processor elements attached to different memory and network elements. Supplementing controllers with additional processors and dispersing tasks results in a more graceful degradation of the FTPP.

The choice of cluster topology not only affects system reliability but also system throughput, maintainability, and modularity. Table 4.1-1 depicts the relative rankings of the three topologies as examined for the baseline case of six clusters. A ranking of one represents the most desirable system, and a ranking of three represents the least desirable. It is left to the system designer to determine the relative weight of each attribute. In the baseline case, the singly linked system is the most reliable followed closely by the fully linked system, and then by the centrally linked system. As the number of clusters in the system increases, the singly linked system stays most reliable, but with an increasing margin over the second best, while the fully and centrally linked systems switch relative positions. The number of clusters, at which the centrally linked system becomes more reliable than the fully linked system, increases as the component failure rates decrease. The addition of a redundant cluster to the baseline case favors the fully linked system over the singly linked system.

	Centrally Linked	Fully Linked	Singly Linked
Throughput	2	1	3
Maintainability	1	3	2
Modularity	1	3	2
Reliability	3	2	1

Table 4.1-1: Relative Rankings of Topologies for Six Cluster System.

Different reliability analysis techniques are more suitable for specific topologies. The assumption of cluster independence can be made for the fully linked system with little loss in the accuracy of the reliability calculations. Using the assumption of cluster independence, the reliability of a fully linked system can be easily estimated for any arbitrary system of n clusters with x redundant clusters. The reliability of a centrally linked system can be easily calculated for a system of n clusters by decomposing on the input/output elements of the central cluster to find the reliability of the system of input/output elements.

This reliability together with an estimate of the cluster's reliability (without input/output elements) are used to estimate the system's reliability. The singly linked system is the most difficult to analyze, but can be accomplished by using conditional probability, since only adjacent cluster links are dependent. Like the centrally linked system, the reliability of the system of input/output elements together with an estimate of the cluster's reliability can be used to estimate the system's reliability. The same methods can be used to calculate the reliability of a centrally or singly linked system with redundant clusters. These calculations, however, are not suited to a system with more than a few redundant clusters due to the increased complexity of the calculations.

Additional clusters in an FTPP system may be used in two ways: to decrease the taskload per cluster, or to serve as spares. Using additional clusters to decrease the tasks per cluster is a passive method which requires only an initial assignment of tasks, while using additional clusters as spares is an active method which requires the migration of the tasks of one cluster to another. From the viewpoint of hardware, reliability modelling shows redundant clusters to be clearly more effective in increasing system reliability. On the other hand, from the viewpoint of software, redundant clusters require more complex programs thus decreasing software reliability. There is a limit to the number of redundant clusters that can be added to a system and still increase system reliability in a fully linked system, but this limit is high enough that it should not be a concern in a practical system design.

The application reliability and throughput requirements are extremely stringent. The application reliability requirement can not be met without using redundant clusters or components whose failure rates are unattainable in the foreseeable future. Using lower bound assumptions, the reliability requirement for a simplex computer can be met with the addition of two redundant clusters to the baseline system using components with MTTFs under 10^5 hours (10^6 hours for network elements). Using upper bound assumptions, the reliability requirement for a triplex computer system can be met with the same redundancy level and component MTTFs. The application throughput requirement can be met relatively easily by using twenty sets of 5 MIPS processors if ideal speedup assumptions are made. Using the speedup bounds of Hwang and Briggs would require between ninety and a million plus sets of processors. The efficiency in the partitioning of software for use in a parallel system is clearly important and may have a profound effect on the system architecture and performance.

4.2 RECOMMENDATIONS FOR FURTHER RESEARCH

1. The combinatorial models used in this thesis are unable to model simultaneous failures. The effect of a component failing during a reconfiguration can be represented using Markov Modelling. The number of states required to model the FTPP, however, would be prohibitively large and simplifications would have to be made.

2. A simpler way is needed to calculate or estimate the reliability of non-fully connected topologies of systems with redundant clusters.

3. This thesis has modelled three basic FTPP topologies; modelling 'hybrid' topologies, which combine the advantages and disadvantages of these basic topologies, would provide the FTPP designer with further options.

4. This thesis has neglected the reliability effects of software. Techniques for computing software reliability are less advanced than those of computer hardware reliability, yet both are equally important in the calculation of a system's reliability.

LIST OF REFERENCES

1. Harper, R. and J. Lala, "A Fault-Tolerant Parallel Processor," Charles Stark Draper Laboratory Internal Report, 5 January 1985.

2. Harper, R., "Fault Tolerant Parallel Processor," Charles Stark Draper Laboratory Independent Research and Development Report, July 1986.

3. Hwang, K. and F.A. Briggs, Computer Architecture and Parallel Processing, McGraw-Hill, 1984.

4. Deyst, J., et. al., "Fault Tolerant Parallel Processor," Charles Stark Draper Laboratory Briefing Package, June 1985.

5. Harper R., Doctoral Thesis in Progress, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 1987 (est.).

6. Adams, M., et. al., "Fault Tolerant Control Systems," Massachusetts Institute of Technology 16.321 Course Notes, February 1983.

7. Maekawa, M., "Optimal Processor Interconnection Topologies," The 8th Annual Symposium on Computer Architecture, Minneapolis, MN, 12-14 May 1981.

8. Levitt, K.N., P.M. Melliar-Smith and R.L. Schwartz, "Fault Tolerant Architectures for Integrated Aircraft Electronics Systems," NASA Contractor Report 172282, SRI International, Menlo Park, CA, June 1985.

9. Finkel, R.A. and M.H. Solomon, "Processor Interconnection Strategies," IEEE Transactions on Computers, Volume C-29, Number 5, May 1980.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/CI/NR 87- 51T

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Reliability Modelling For A Fault-Tolerant Parallel Processor

5. TYPE OF REPORT & PERIOD COVERED
THESIS/~~DISSERTATION~~

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Edward I. Gjermundsen

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
AFIT STUDENT AT:
Massachusetts Institute of Technology

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
AFIT/NR
WPAFB OH 45433-6583

12. REPORT DATE
January 1987

13. NUMBER OF PAGES
103

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver

LYNN E. WOLAVER 17 Aug 87
Dean for Research and
Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
ATTACHED

87 10 14 250

END
DATE
FILMED
JAN
1988